



Global Knowledge®

DEVOPS

Version 2018

Présentation

- Qui suis-je ?



Présentation

- Qui êtes-vous ?



Logistique



- Pause en milieu de session



- Vos questions sont les bienvenues. N'hésitez pas !



- Feuille d'évaluation à remettre remplie en fin de session



- Merci d'éteindre vos téléphones

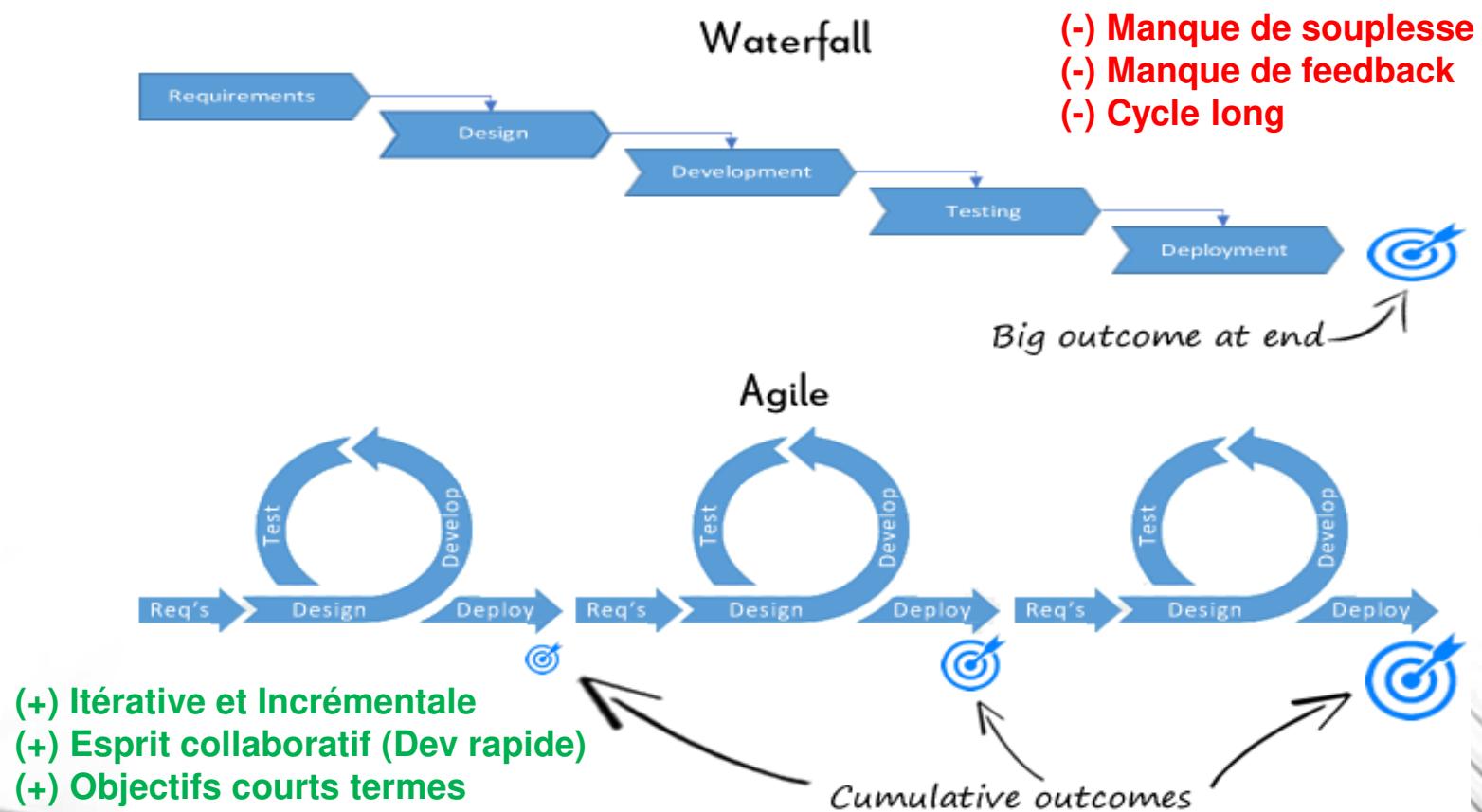
Objectifs

- Introduire la méthodologie DevOps et ses bonnes pratiques pour le développement logiciel moderne.
- Connaitre la chaîne logicielle pour mettre en place un environnement DevOps (déployé en local ou bien hébergé sur le Cloud)
 - d'Intégration Continue,
 - de Livraison Continue
 - de Déploiement Continu

Sommaire

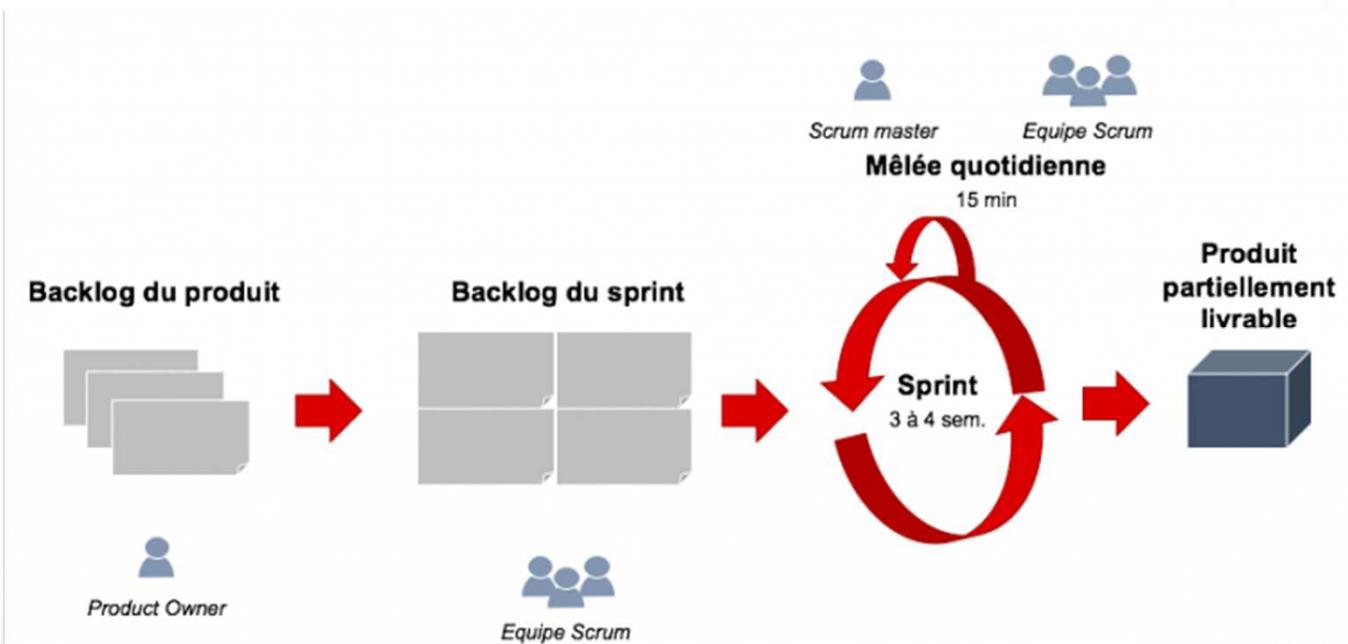
- **Chapitre 1 : Introduction**
- **Chapitre 2 : L'intégration Continue**
- **Chapitre 3 : Automatisation des Builds**
- **Chapitre 4 : Le gestionnaire de sources**
- **Chapitre 5 : Le serveur d'intégration continue : Jenkins**
- **Chapitre 6 : La gestion des livrables**

Introduction



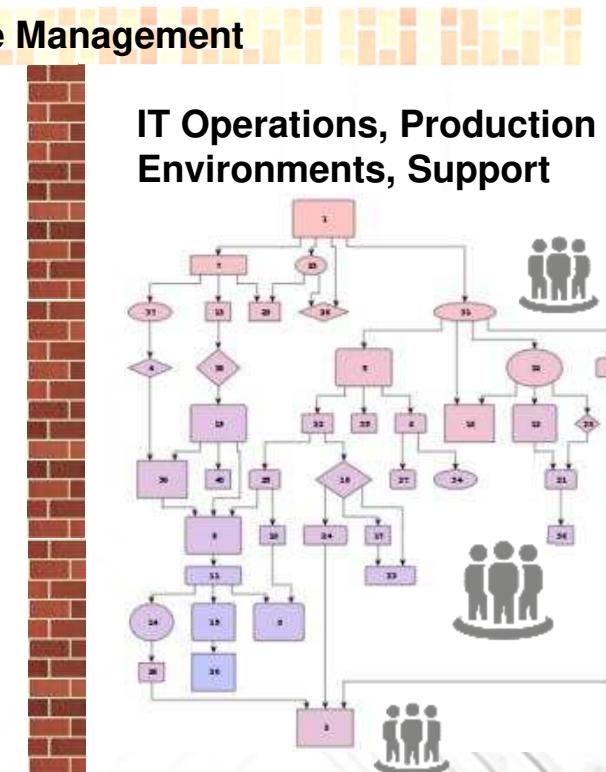
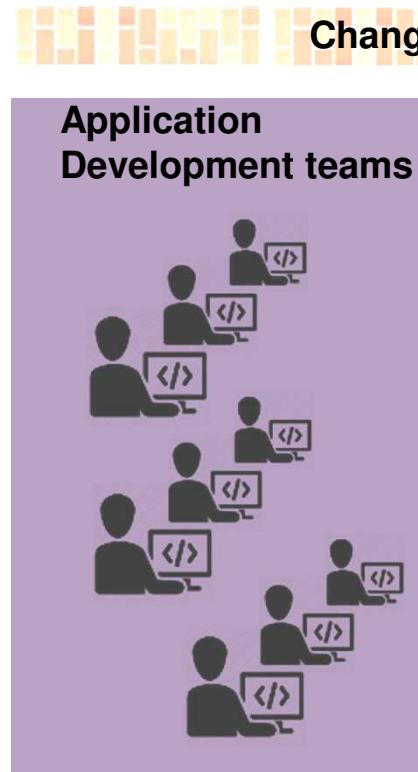
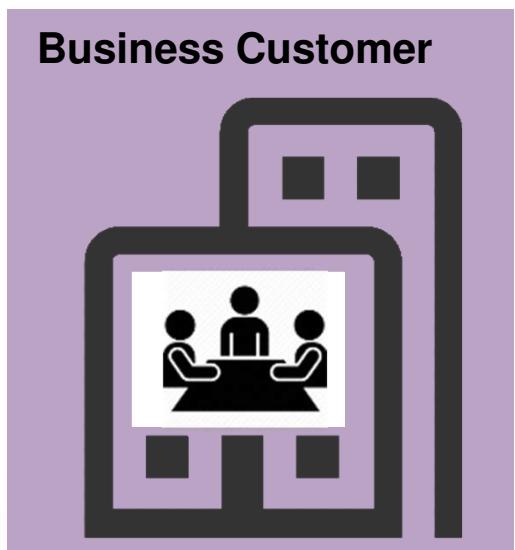
Introduction : Méthodes agiles : Scrum

- Améliorer la **productivité** des équipes
- Le processus de développement s'articule en effet autour **d'une équipe soudée**, qui cherche à atteindre un même but.



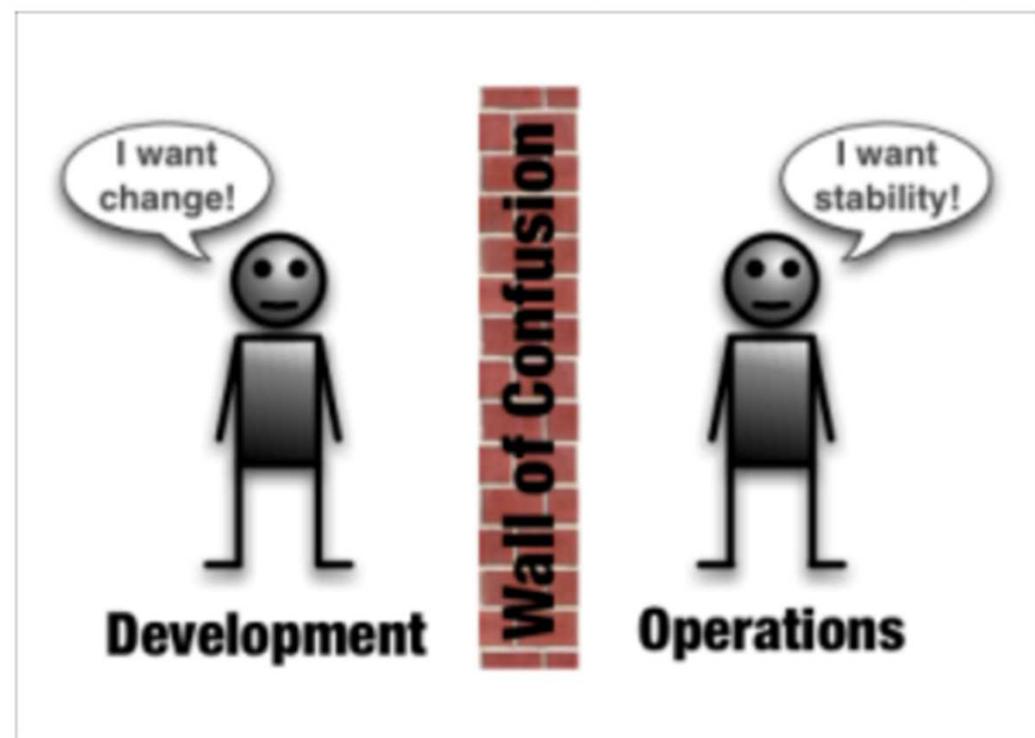
Introduction

- Vision simplifiée de l'entreprise



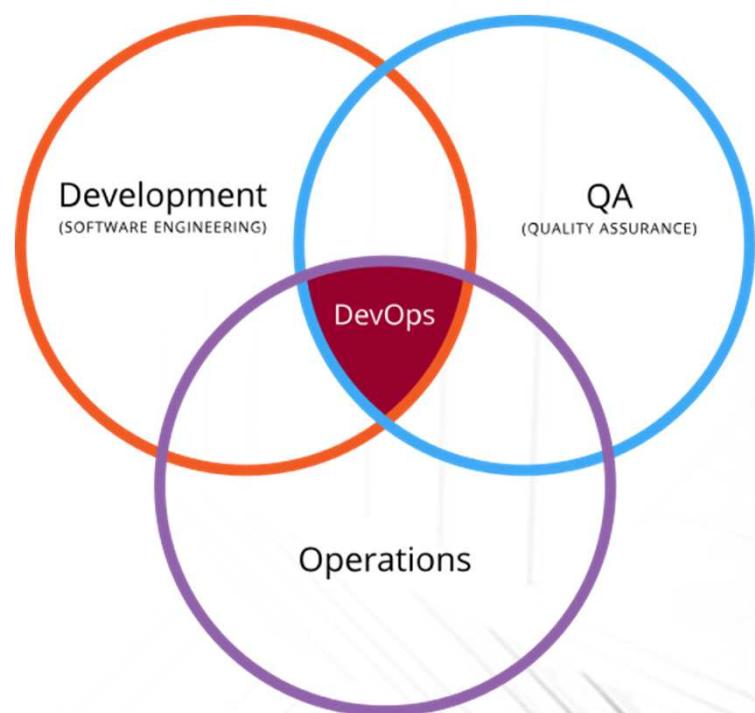
Introduction : Entreprise : Clash DEV / OPS en vue

- Comment faire tomber les murs ?
- Améliorer la capacité d'une entreprise à **livrer des applications et des services** à un **rythme élevé**.



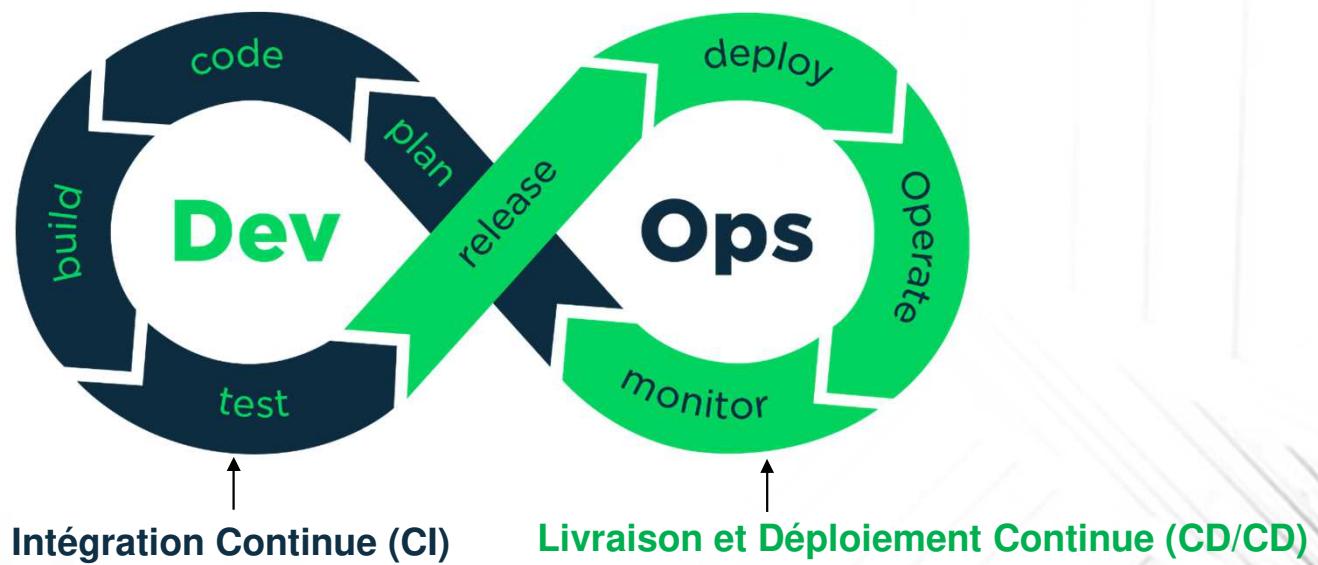
Introduction : Une nouvelle mentalité : DevOps Mindset

- Établir une culture et un environnement où les logiciels de construction, de test et de diffusion peuvent se produire rapidement, fréquemment et de manière plus fiable
- Réduire le Time To Market
- Permettre aux entreprises de mieux servir leurs clients et de gagner en compétitivité

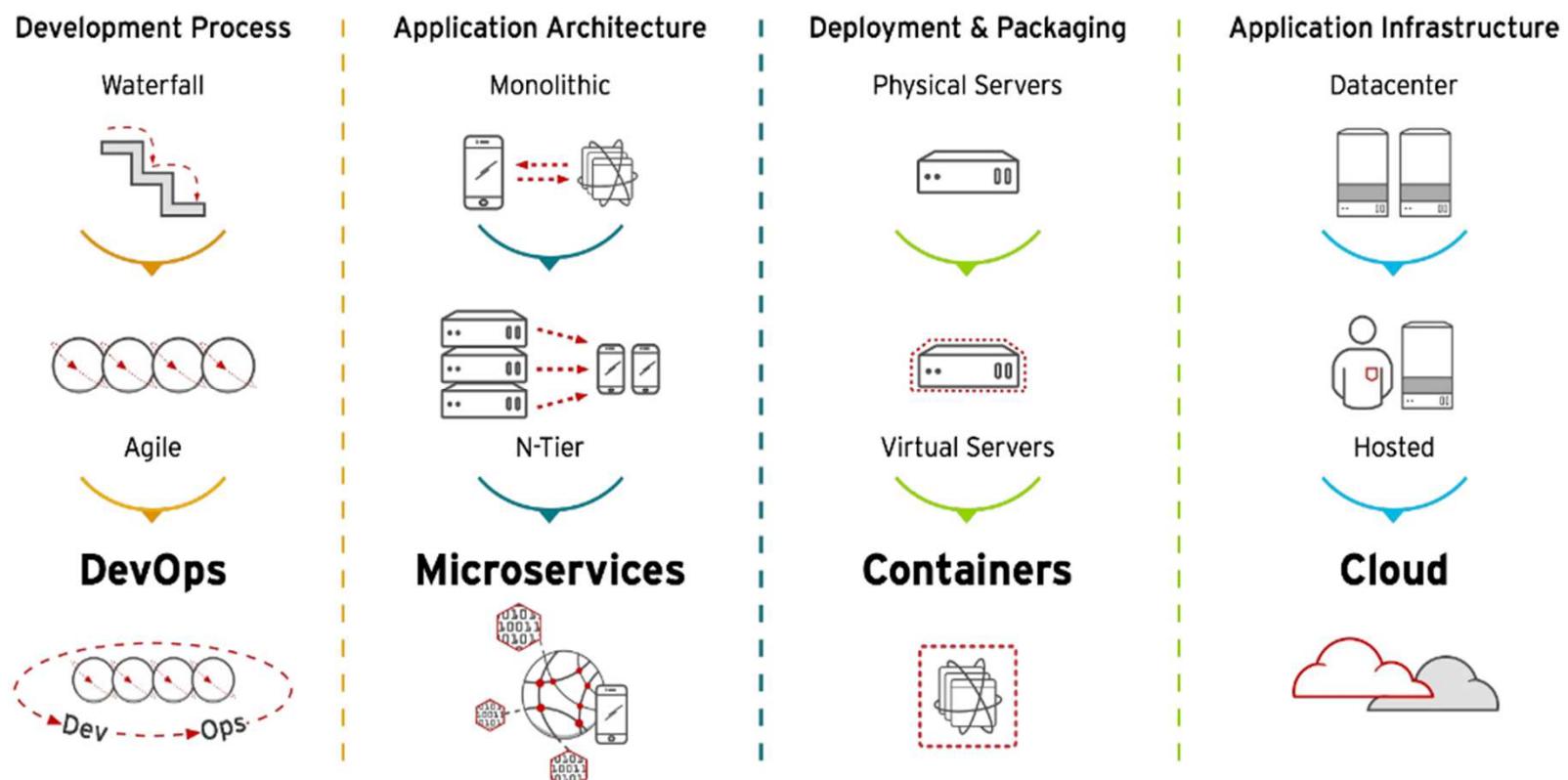


Introduction : DevOps Toolchain : Agile + Automation and Continuous

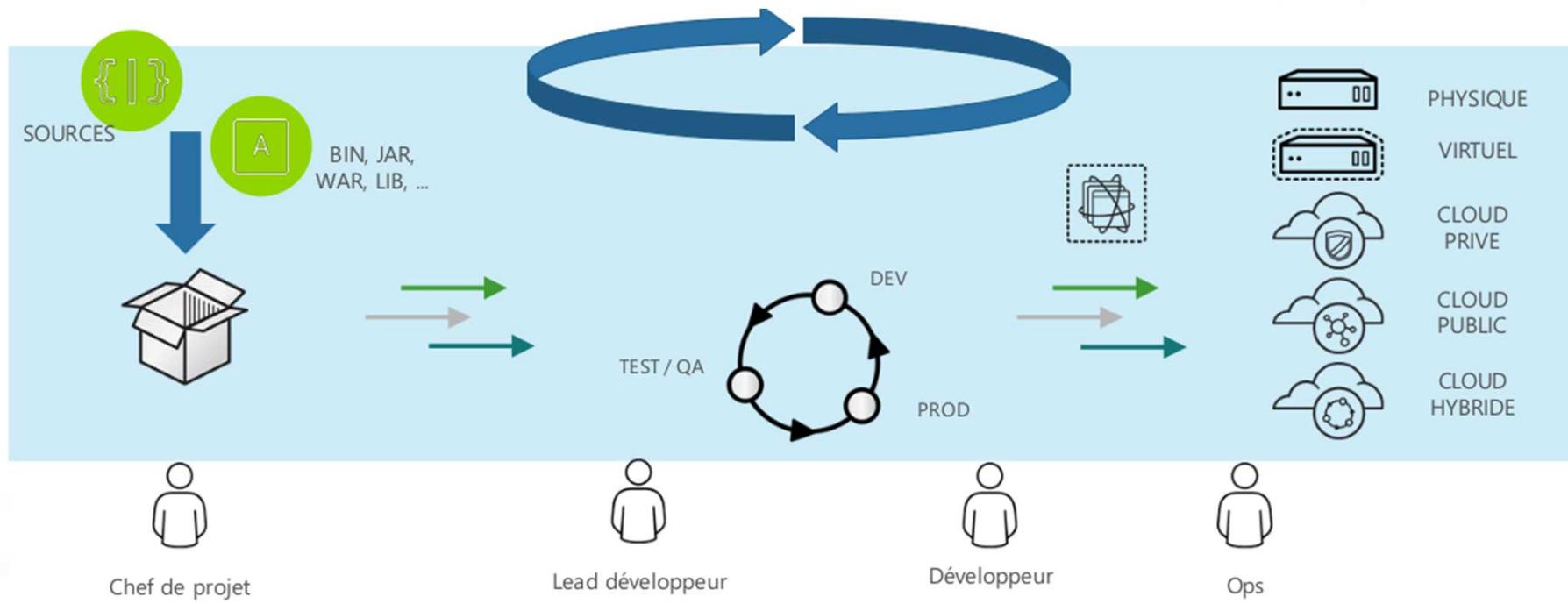
- Étend et complète le processus d'intégration et de diffusion continue en s'assurant que le code est prêt à la production (**CI**)
- DevOps permet un flux de travail beaucoup plus continu dans les opérations IT (**CD**)



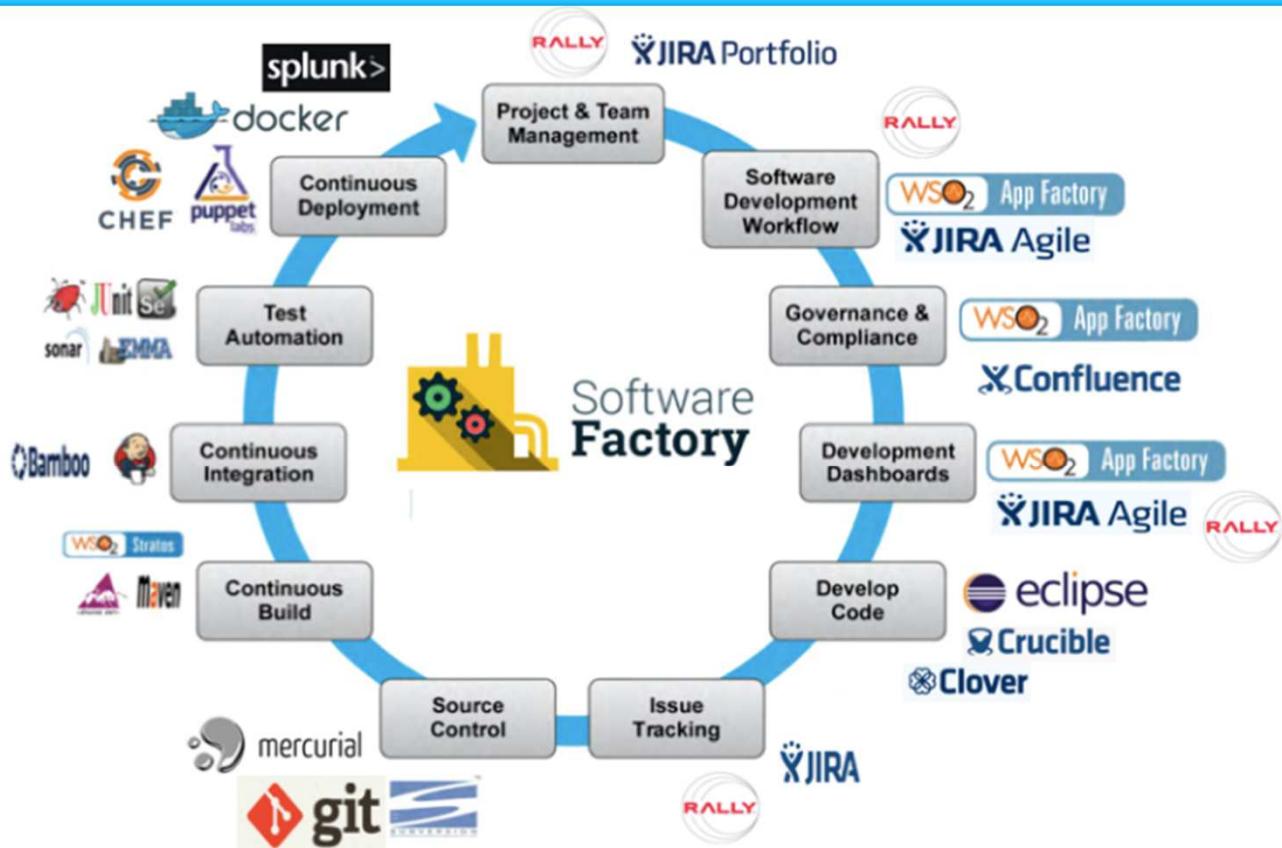
Introduction : DevOps : Positionnement par rapport au Modern IT



Introduction : DevOps : Usine logicielle (Accélération du Time To Market)

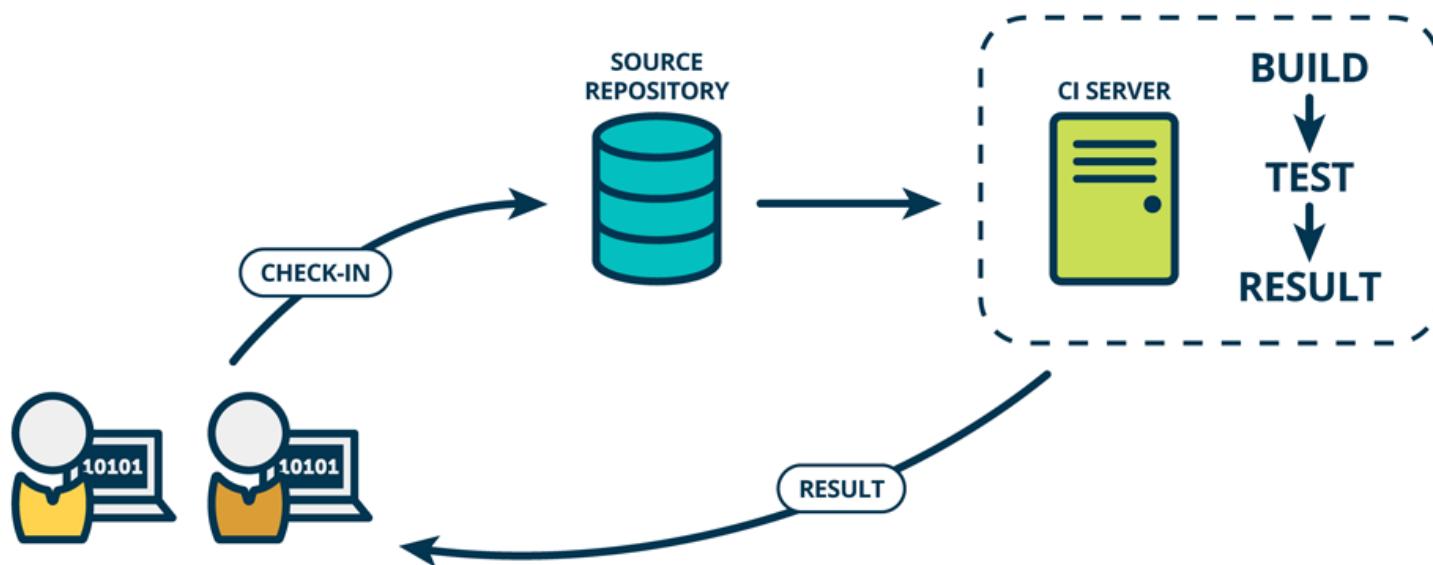


Introduction : DevOps Toolchain : Panoplie d'outils



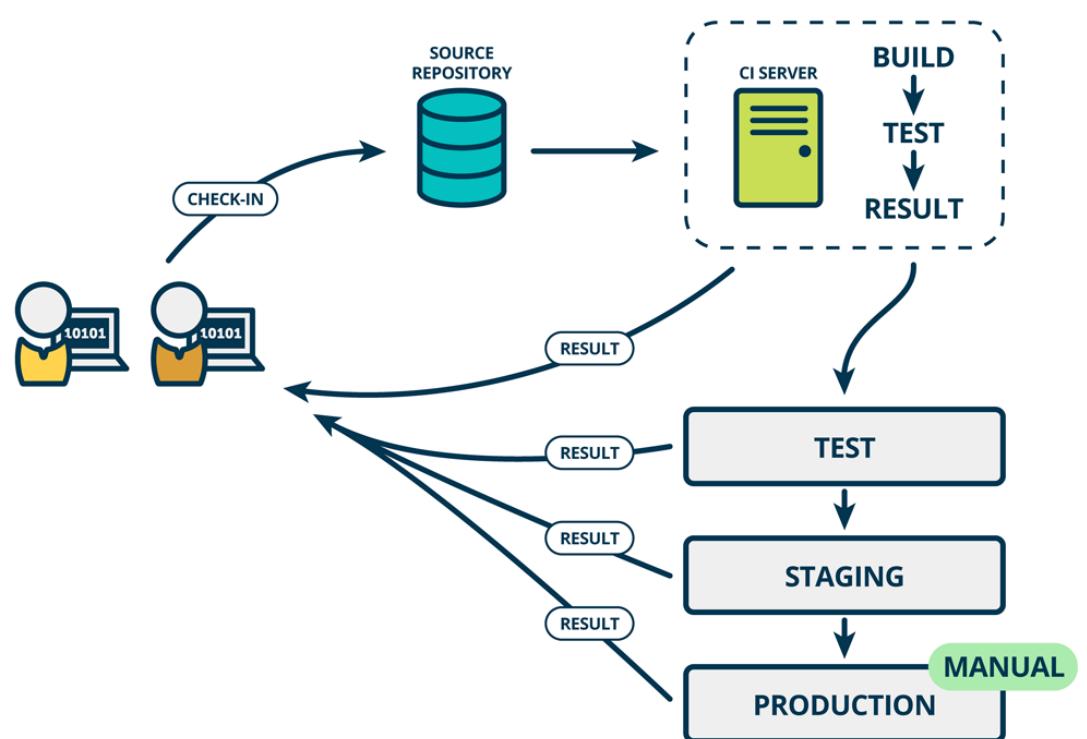
Introduction : Démarche DevOps : Intégration Continue CI

- Méthode de développement qui permet **d'intégrer régulièrement** les modifications de code à un référentiel centralisé. Chaque intégration est validée par une **construction logiciel automatique**



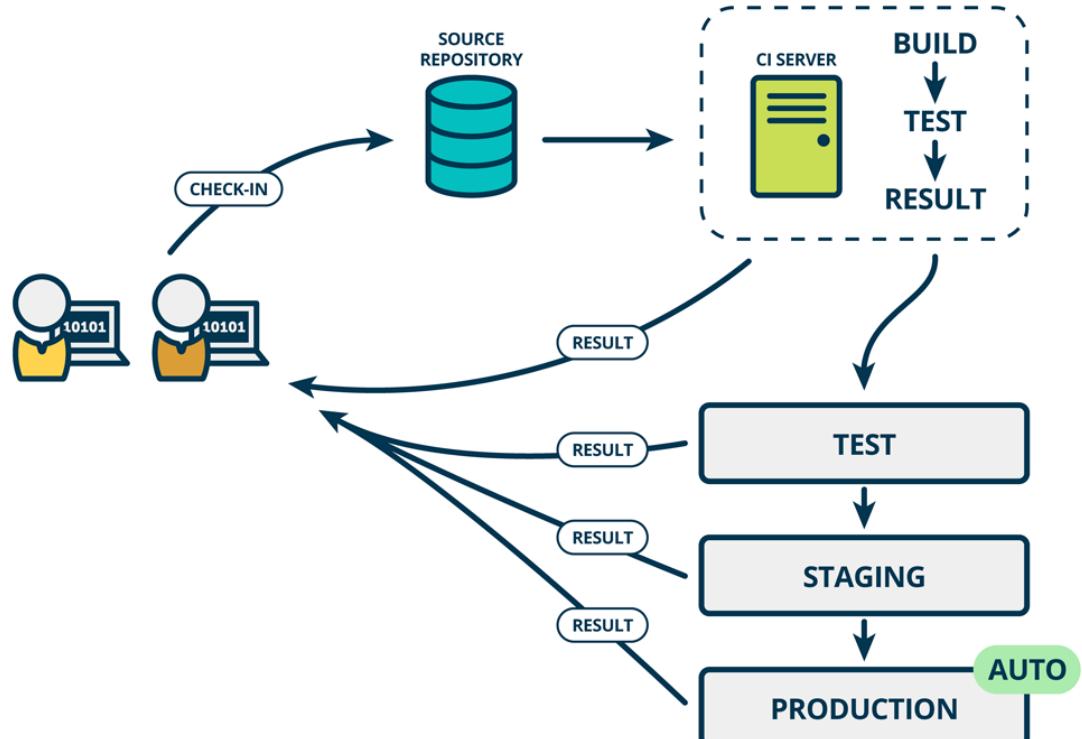
Introduction : Démarche DevOps : Livraison Continue CD

- Une technique qui consiste à tester, préparer et déployer un changement de code.
- Une validation humaine finale sera à réaliser avant le déploiement final.

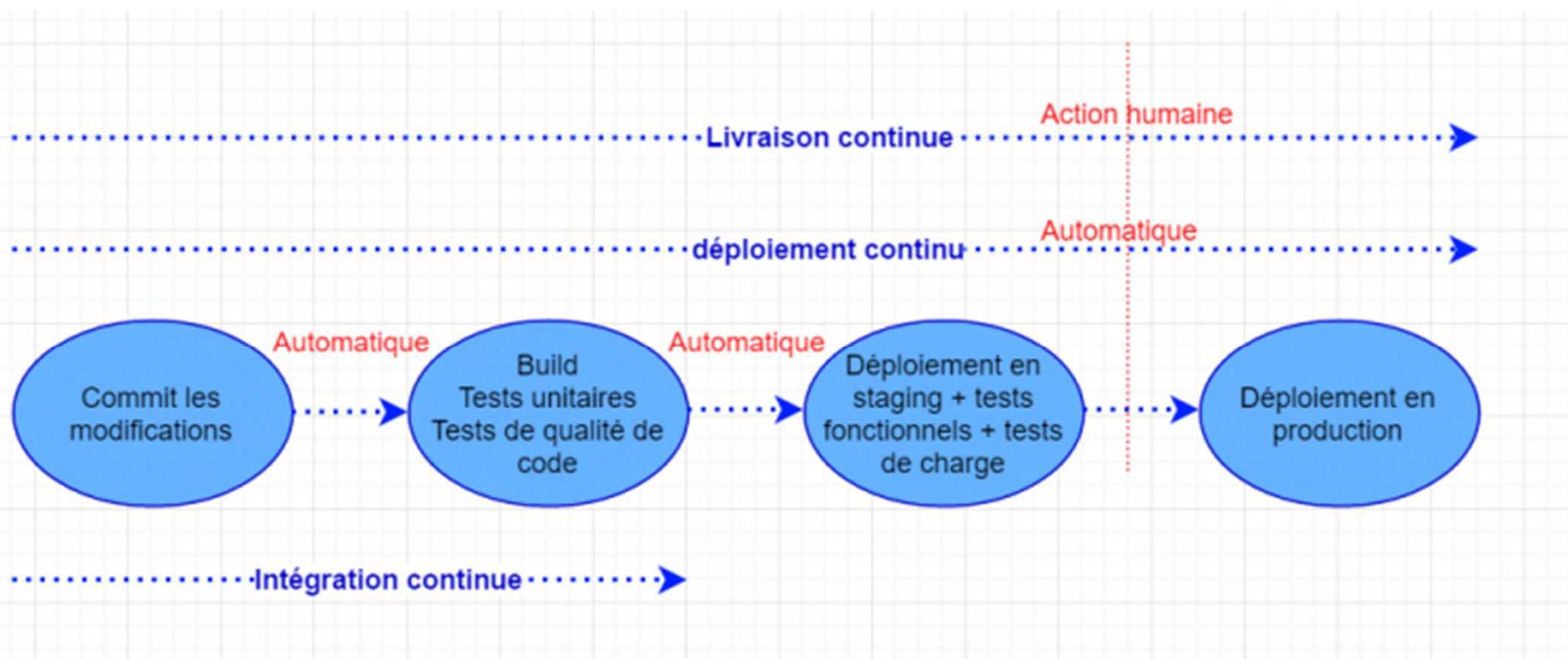


Introduction : Démarche DevOps : Déploiement Continue CD

- Une technique d'ingénierie informatique similaire au Livraison Continue sauf que le déploiement s'automatise sans validation en amont.



Introduction : Démarche DevOps : CI vs CD vs CD



Questions & Réponses

Q&R

Sommaire

- **Chapitre 1 : Introduction**
- **Chapitre 2 : L'intégration Continue**
- **Chapitre 3 : Automatisation des Builds**
- **Chapitre 4 : Le gestionnaire de sources**
- **Chapitre 5 : Le serveur d'intégration continue : Jenkins**
- **Chapitre 6 : La gestion des livrables**

Introduction à l'intégration continue

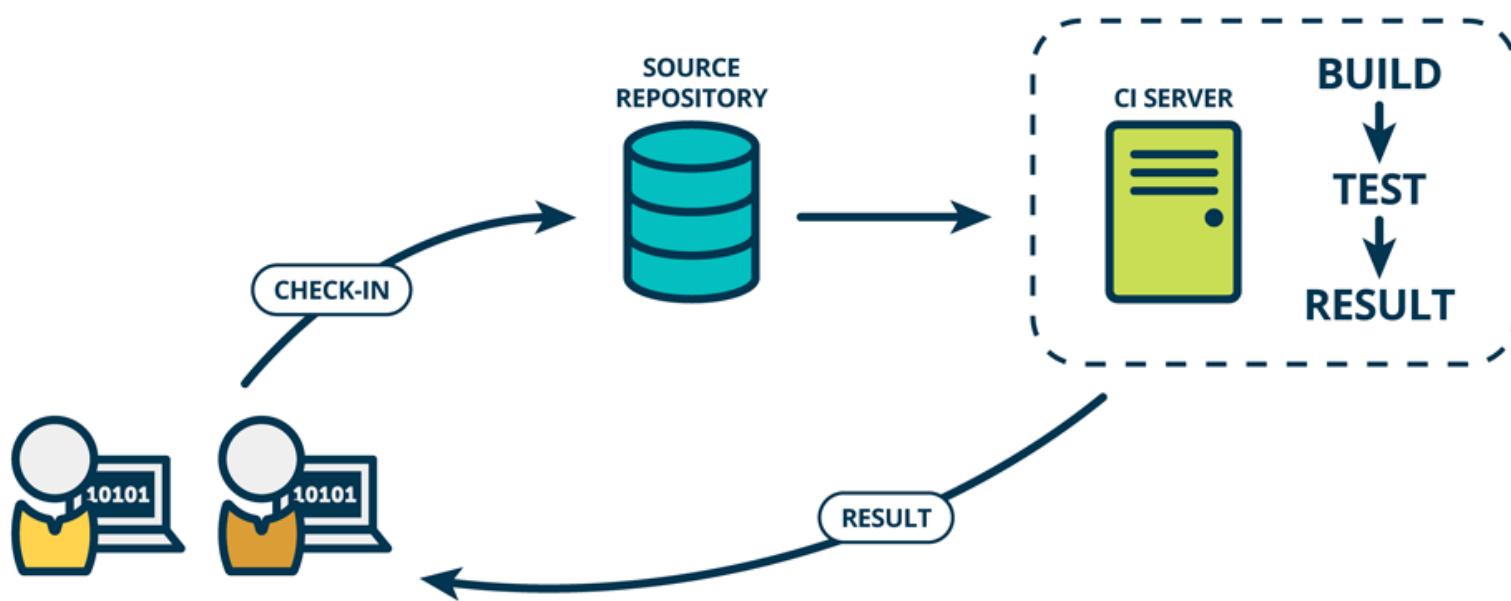
- Agenda
 - [Introduction](#)
 - Définition de l'intégration continue
 - Les principes de l'intégration continue
 - Les pré-requis de l'intégration continue
 - Les bénéfices de l'intégration continue
 - Les différents outils de l'intégration continue
 - Mise en place de l'environnement d'intégration continue

Définition de l'intégration continue

- « ... une pratique de développement logiciel où les membres d'une équipe intègrent leur travail fréquemment, habituellement chacun au moins une fois par jour – ce qui entraîne plusieurs intégrations par jour. Chaque intégration est validée par une construction, ‘build’, automatique (ce qui inclut les tests) pour détecter les erreurs d'intégration aussi vite que possible ... »
 - Martin Fowler

<http://www.martinfowler.com/articles/continuousIntegration.html>

Workflow de l'intégration continue



Introduction à l'intégration continue

- Agenda
 - Introduction
 - Définition de l'intégration continue
 - **Les principes de l'intégration continue**
 - Les pré-requis de l'intégration continue
 - Les bénéfices de l'intégration continue
 - Les différents outils de l'intégration continue
 - Mise en place de l'environnement d'intégration continue

Les principes de l'intégration continue

- Gestion centralisée du code source
 - Maintenir un dépôt unique de code source.
 - Le code source est partagé entre les membres de l'équipe via un outil de gestion de source.

Les principes de l'intégration continue

- Construction automatique

- La construction englobe l'ensemble des actions souhaitées prenant en entrée des fichiers sources pour produire un résultat souhaité.
- La construction doit être faite de sorte qu'elle soit **automatisable**.
- L'ensemble des étapes de la construction doivent être réalisables par un **outil de construction logicielle**.
- L'automatisation de la construction permet de lancer via une commande simple un ensemble de processus complexes qui s'enchainent dans un ordre précis.

Les principes de l'intégration continue

- Construction auto-testante
 - Les **tests** font partie de la construction.
 - Les tests doivent être exécutables dans un mode **automatique** pour qu'ils puissent être pris en compte lors de la construction du logiciel.
 - Les tests permettent de valider le bon fonctionnement du livrable produit par la construction.
 - **L'échec d'un test entraîne l'échec de la construction.**
 - Les tests doivent être passant à 100%.

Les principes de l'intégration continue

- Construction de courte durée
 - La **durée** de la construction détermine la fréquence d'exécution de la construction.
 - Plus la construction est **rapide** plus il sera possible de **lancer des intégrations** dans la journée et de savoir :
 - Si le code source compile toujours.
 - Si les tests sont toujours passants.
 - Si tous les composants du logiciel fonctionnent correctement ensemble.
 - Si le logiciel se déploie toujours correctement, etc.
 - En cas de problème détecté il sera plus facile de corriger les anomalies.

Les principes de l'intégration continue

- Toute modification du logiciel est intégrée
 - Toute modification d'objet de configuration du logiciel **doit être intégrée et testée** sur la machine d'intégration continue pour valider son impact sur le livrable.

Les principes de l'intégration continue

- Environnement d'intégration ISO production
 - L'environnement de test du livrable produit par la construction doit être le plus proche de l'environnement de production (ISO production).
 - Les écarts entre **l'environnement de test et l'environnement de production** peuvent conduire à **un échec de déploiement** du logiciel **en production** même si le déploiement et les tests étaient concluant en environnement d'intégration.
 - Architecture matérielle.
 - Version de système d'exploitation.
 - Version de machine virtuelle.
 - Etc.

Les principes de l'intégration continue

- Tout le monde doit être au courant de ce qui se passe
 - Le déroulement des processus de construction, de test et de déploiement doit être visible à tous les contributeurs du projet.

Les principes de l'intégration continue

- Synthèse des principes
 - Gestion centralisée du code source.
 - Construction automatique.
 - Construction auto-testante.
 - Construction de courte durée.
 - Toute modification du logiciel est intégrée.
 - Environnement d'intégration ISO production.
 - Tout le monde doit être au courant de ce qui se passe.

Introduction à l'intégration continue

- Agenda
 - Introduction
 - Définition de l'intégration continue
 - Les principes de l'intégration continue
 - **Les pré-requis de l'intégration continue**
 - **Les bénéfices de l'intégration continue**
 - **Les différents outils de l'intégration continue**
 - Mise en place de l'environnement d'intégration continue

Les prérequis de l'intégration continue

- Publication régulière du code source
 - Plus la quantité de code publiée est **grosse** plus le risque de **conflit** augmente.
 - La publication régulière du code source :
 - Réduit le risque de conflit.
 - Facilite l'identification de la modification à l'origine du problème.
 - Pour pouvoir publier régulièrement le code source il faut :
 - Publier les modifications apportées à **la fin de chaque tâche** de développement.
 - Faire **des petits changements** et ne pas apporter des changements à plusieurs composants dans une seule tâche.

Les prérequis de l'intégration continue

- Publication régulière du code source
 - Chaque développeur **doit publier** ses modifications du code source au moins une fois par jour.

Les prérequis de l'intégration continue

- Exécuter des constructions privées
 - Pour éviter les **constructions cassées**, le développeur doit exécuter **des constructions privées** sur son poste de travail.
 - Permet de **déetecter et corriger les erreurs de compilation** avant la publication du code source modifié par le développeur.
 - Permet de **déTECTER ET CORRIGER LES TESTS EN ÉCHEC** avant la publication du code source modifié.

Les prérequis de l'intégration continue

- Ne pas publier du code qui ne compile pas
 - La publication de code qui ne compile pas implique systématique l'échec de la construction.
 - L'usine logicielle est alors mise hors service.

Les prérequis de l'intégration continue

- Réparer les constructions cassées immédiatement
 - Une construction est dite cassée en cas de problème empêchant son succès :
 - Erreur de compilation.
 - Test en échec.
 - Problème de déploiement.
 - Etc.
 - La réparation des constructions cassées est une urgence pour le projet.
 - Le responsable du commit à l'origine de l'échec de la construction porte la responsabilité de la réparation.

Les prérequis de l'intégration continue

- Synthèse des prérequis
 - Publication régulière du code source.
 - Exécuter des constructions privées.
 - Ne pas publier du code qui ne compile pas.
 - Réparer les constructions cassées immédiatement.

Introduction à l'intégration continue

- Agenda
 - Introduction
 - Définition de l'intégration continue
 - Les principes de l'intégration continue
 - Les pré-requis de l'intégration continue
 - **Les bénéfices de l'intégration continue**
 - Les différents outils de l'intégration continue
 - Mise en place de l'environnement d'intégration continue

Les bénéfices de l'intégration continue

- Réduire les risques – Détection tardive des anomalies
 - Les anomalies sont détectées et corrigées au plus tôt :
 - Les anomalies ont plus de chance d'être détectées au moment de leur introduction dans le code source.
 - Réduit le risque de la détection tardive des anomalies.

Les bénéfices de l'intégration continue

- Réduire les risques – Non qualité
 - La qualité logicielle est **mesurable** via des **métriques** actualisées en continu :
 - Suit à l'exécution des tests automatiques et de l'inspection du code source.
 - Réduit le risque de la non qualité logicielle.

Les bénéfices de l'intégration continue

- Réduire les risques – Logiciel non déployable
 - Le logiciel est construit et déployé dans un environnement à l'image de son environnement de production :
 - La machine d'intégration continue est une référence.
 - Réduit les risques de la détection tardive des dépendances par rapport au système d'exploitation, variables d'environnement, les options de compilation, librairies tierces, etc.
 - Le livrable du logiciel est **reproductible dans le temps et dans l'espace**.
 - **Je ne comprends pas, ça marche sur mon poste !?**



- Automatiser les tâches manuelles répétitives

Les bénéfices de l'intégration continue

- Garantit que :
 - Les processus sont toujours exécutés de la même manière.
 - Les processus sont toujours exécutés dans **le même ordre**.
 - Les processus sont exécutés **à chaque changement du code**.
- Assure un gain de temps, d'effort et de budget.
- Permet à l'équipe de se concentrer sur des tâches à forte valeur ajoutée.

Les bénéfices de l'intégration continue

- Générer des versions intermédiaires du logiciel
 - Une version intermédiaire du logiciel est générée à chaque intégration :
 - Une version **déployable** et **testable** du produit peut être générée à tout moment.
 - Améliore **la confiance de l'équipe** de développement par rapport aux changements apportés.
 - Améliore **la confiance du client** par rapport à la pertinence des réalisés.
 - Permet s'immuniser contre l'effet tunnel, première cause d'échec des projets informatique.

Les bénéfices de l'intégration continue

- Améliorer la visibilité projet
 - Prise de décision efficace :
 - La prise de décision se fait sur la base d'informations actualisées en continu et partagées dans l'équipe de développement.
 - Mesure des tendances :
 - Observation de l'évolution des indicateurs de la qualité logicielle.
 - Constructions réussies.
 - Tests passants.
 - Couverture des tests.
 - Etc.

Les bénéfices de l'intégration continue

- Synthèse des bénéfices
 - Réduire les risques – Détection tardive des anomalies.
 - Réduire les risques – Non qualité.
 - Réduire les risques – Logiciel non déployable.
 - Automatiser les tâches manuelles répétitives.
 - Générer des versions intermédiaires du logiciel.
 - Améliorer la visibilité projet.

Introduction à l'intégration continue

- Agenda
 - Introduction
 - Définition de l'intégration continue
 - Les principes de l'intégration continue
 - Les pré-requis de l'intégration continue
 - Les bénéfices de l'intégration continue
 - **Les différents outils de l'intégration continue**
 - Mise en place de l'environnement d'intégration continue

Les outils de l'intégration continue

- **Un atelier de développement :**
 - Facilite la production du code source.
- **Un gestionnaire de sources :**
 - Partager le code source, documentation, etc.
 - Tracer toutes les modifications.
- **Un système de test unitaires :**
 - Contrôler la validité du code source.
 - Déetecter les régressions.
 - Peut être lancé en mode automatique ou en mode manuel.

Les outils de l'intégration continue

- **Un outil de construction:**
 - Gérer les dépendances.
 - Automatiser la génération de l'exécutable à partir du code source.
- **Un serveur d'intégration continue :**
 - Surveiller les changements apportés au code source.
 - Exécuter une construction d'intégration à chaque changement du code source.
- **Un outil de surveillance de la qualité du code :**
 - Inspecter le code source.
 - Établir des métriques sur la complexité du code, le taux de commentaire, le code dupliqué, etc.

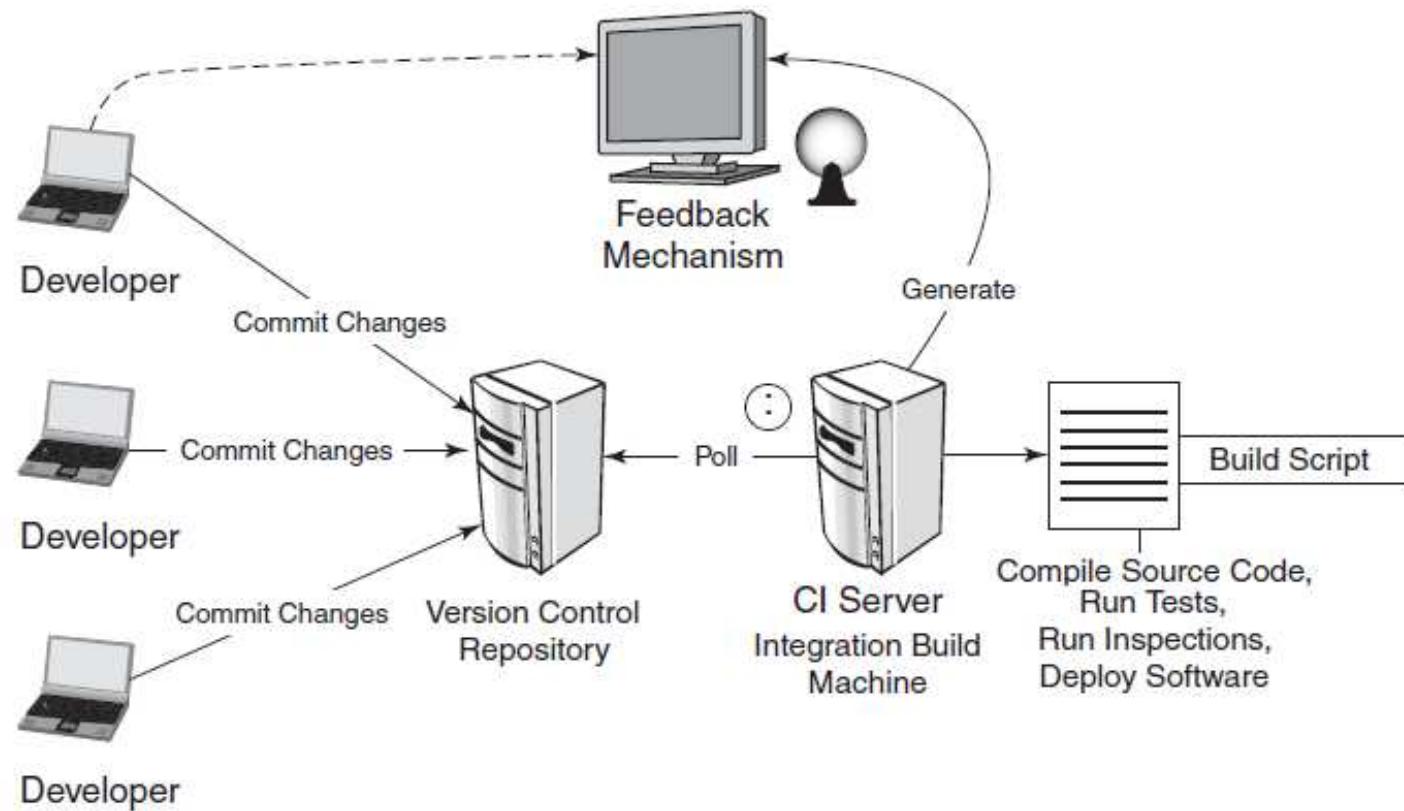
Les outils de l'intégration continue

- **Un repository manager :**
 - Gérer les composants binaires.
 - Les dépendances du projet
 - Les artefacts produits par le projet.
- **Un outil de communication :**
 - Informer en temps réel de l'état de la construction d'intégration.
- D'autres outils : Monitoring (Nagios), Logs (Kibana), etc.

Introduction à l'intégration continue

- Agenda
 - Introduction
 - Définition de l'intégration continue
 - Les principes de l'intégration continue
 - Les pré-requis de l'intégration continue
 - Les bénéfices de l'intégration continue
 - Les différents outils de l'intégration continue
 - **Mise en place de l'environnement d'intégration continue**

Environnement d'intégration continue



Quiz

QUIZ

55

Quiz

1. L'intégration continue est un processus :

- a. Agile.
- b. Couteux.
- c. Trop compliqué à mettre en œuvre.

2. Le processus d'intégration intervient dans :

- a. Le recueil des besoins client.
- b. La conception du produit logiciel.
- c. La construction et la livraison du produit logiciel.

3. L'intégration continue permet de :

- a. Produire un logiciel ne comportant aucune anomalie.
- b. Réduire le risque de la détection tardive des anomalies.
- c. Optimiser le coût de la correction des anomalies.

Quiz

4. N'est pas un principe de l'intégration continue :

- a. Gestion centralisée du code source.
- b. Construction automatique.
- c. Toute modification du logiciel est intégrée.
- d. La publication régulière des modifications du code source.

5. N'est pas un prérequis de l'intégration continue :

- a. Publication régulière du code source.
- b. Disposer d'un environnement de construction privée ISO production.
- c. Exécuter des constructions privées.
- d. Ne pas publier du code qui ne compile pas.
- e. Réparer les constructions cassées immédiatement.

Questions & Réponses

Q&R

58

Sommaire

- **Chapitre 1 : Introduction**
- **Chapitre 2 : L'intégration Continue**
- **Chapitre 3 : Automatisation des Builds**
- **Chapitre 4 : Le gestionnaire de sources**
- **Chapitre 5 : Le serveur d'intégration continue : Jenkins**
- **Chapitre 6 : La gestion des livrables**

L'automatisation des builds

- Agenda

- Qu'est ce que la construction d'un logiciel
- Objectifs des outils de construction
- Prérequis d'une construction automatisable
- Différents types de constructions
- Construction logicielle avec Maven

Qu'est ce que la construction d'un logiciel

- Dans le contexte du développement logiciel, la construction d'un logiciel correspond aux processus qui convertit le code source du projet ainsi que tout autre élément de configuration sous la responsabilité du développeur dans un logiciel sous sa forme finale.
- Comporte :
 - La compilation du code source.
 - Le packaging des fichiers compilés dans un format compressé tel que jar, zip, war, etc.
 - La production des installateurs.
 - Etc...
- Terme apparu avec la création de l'outil make pour les systèmes Unix en 1977.

Objectifs des outils de construction

- Simplifier le processus de construction.
- Accélérer la compilation du code source.
- Améliorer la qualité du produit final.
- Fournir un système de construction uniforme:
 - La maîtrise de la construction d'un projet avec un outil donné permet de faciliter la maîtrise des constructions des projets utilisant le même outil.

Prérequis d'une construction automatisable

- Une construction logicielle est dite automatique quand toutes ses étapes :
 - Sont répétables.
 - Ne nécessitent aucune intervention humaine.
 - Peut être exécutée à tout moment sans besoin d'information autre que où est stocké le code source du projet.

Les différents types de construction

- Les termes construction et build seront utilisés indifféremment.
- Build: un ensemble d'activités qui permettent de:
 - Compiler le code source.
 - Générer de la documentation.
 - Tester le code source.
 - Inspecter le code source.
 - Déployer le logiciel généré à partir du code source.
- Build privé: un build exécuté sur le poste de travail du développeur avant de partager ses modifications de code source avec l'équipe.
 - Peut se restreindre à un sous ensemble de tâches du build en fonction de la taille du projet.

Les différents types de construction

- Build d'intégration : un build exécuté sur la machine d'intégration continue à chaque nouvelle publication de modification de code source :
 - Peut se restreindre à un sous ensemble de tâches du build en fonction de la taille du projet.
- Build de nuit : un build complet qui exécute toute les tâches prévues dans le processus de build du logiciel.
 - Exécuté sur la machine d'intégration continue en période de baisse d'activité.
- Build de release : Permet de produire une version du logiciel destinée aux utilisateurs finaux :
 - Intègre obligatoirement l'exécution des tests d'acceptation.

Construction logicielle avec Maven / Présentation de Maven

- Un outil de construction d'application Java:
 - Génère une application « déployable » à partir d'un code source.
 - Compile.
 - Exécute des tests.
- Un outil de gestion de développement qui permet de gérer :
 - Documentation & Rapports.
 - Site web, etc.
- Projet Apache :
 - <http://maven.apache.org>
 - Développé en Java



23/08/2018

66

Construction logicielle avec Maven : Les concepts Maven

- Privilégier la standardisation à la liberté, Convention over Configuration :
 - Structure standard des répertoires d'une application
 - Cycle de développement standard d'une application
- Factoriser les efforts :
 - Un dépôt global regroupant les ressources / bibliothèques communes.
 - Des dépôts locaux.
 - Un dépôt personnel.
- Multiplier les possibilités:
 - Une application légère.
 - De nombreux plugins, chargés automatiquement au besoin.

Construction logicielle avec Maven : Vocabulaire Maven

- **Plugin :**
 - Extension de l'application de base, proposant un ensemble de goal.
 - Le noyau de Maven constitue un framework d'exécution de plugin.
- **Goal :**
 - Tâche proposée par un plugin permettant de lancer un certain nombre d'action.
 - Invoqué par mvn plugin:but.
 - Paramétré par -Dparam=valeur.
- **Maven Project Lifecycle:**
 - Détermine clairement comment est construit un artefact.

Construction logicielle avec Maven : Vocabulaire Maven

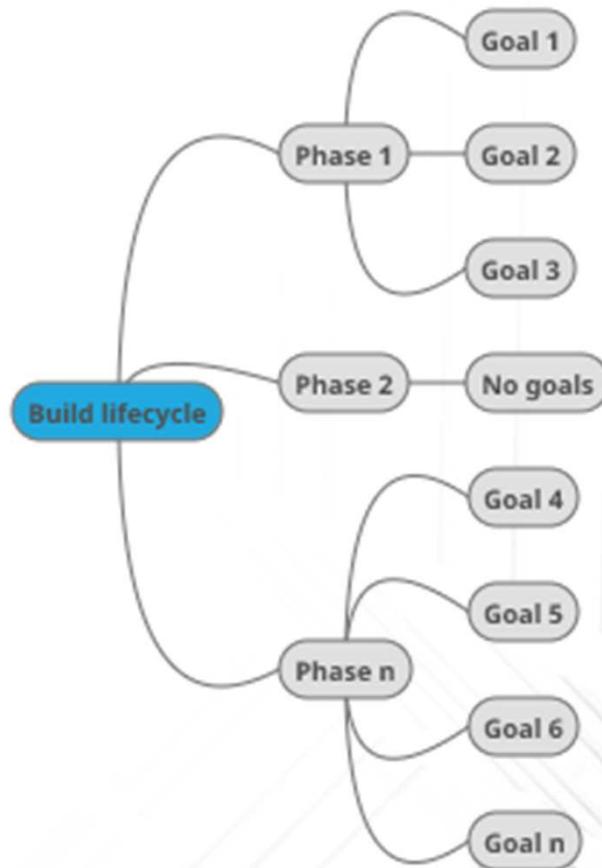
- Une séquence de phases qui permettent d'ordonner des groupes de goals.
- Phase:
 - Étape du cycle de développement d'un logiciel, généralement associée à des buts.
 - Exécutée avec la commande mvn phase.
- Artifact:
 - Application dont le développement est géré via Maven.
- POM:
 - Un projet est décrit par un fichier XML mettant le projet en place.
 - Ce fichier est appelé POM pour Projet Object Model.

Construction logicielle avec Maven : Exemples de plugins et goals Maven

- **Plugin archetype:**
 - Usage: pour créer de nouveaux projets standards.
 - Goals: generate, create.
- **Plugin compiler:**
 - Usage: pour la compilation de code java.
 - Goals: compile.
- **Plugin jar:**
 - Usage: pour générer des artefacts de type jar.
 - Goals: jar, test-jar.
- **Plugin surefire:**
 - Usage: pour l'exécution des tests
 - Goals: test.

Construction logicielle avec Maven : Cycle de vie, phases et goals

- Exécution du cycle de vie:
 - Exécution des phases associées.
- Exécution d'une phase:
 - Exécution des goals associés.
 - Exécution de la phase précédente.
- Exécution d'un goal:
 - Exécution du goal seulement.



Construction logicielle avec Maven : Cycles de vie Maven

- Il existe trois cycles de vie implémentés par Maven.
- Le cycle de vie Clean:
 - Nettoie l'environnement de construction des fichiers générés par la précédente construction.
- Le cycle de vie Default:
 - Traite la construction et le déploiement du projet.
- Le cycle de vie Site:
 - Génère la documentation du projet sous forme d'un site web.
- Chacun de ces cycles de vie consiste en un ensemble de phases exécutées dans ordre spécifique.

Construction logicielle avec Maven : Le cycle de vie Default

- Cycle de vie par défaut permettant de construire un projet java.
- Composé de 23 phases.
 - Validate : valide que le projet est correct et que les informations nécessaires à la construction sont disponibles.
 - Compile : compile le code source du projet.
 - Test : Teste le code source compilé du projet.
 - Package : génère le livrable dans le format attendu.
 - Verify : vérifie les résultats des tests d'intégration.
 - Install : installe le livrable dans un dépôt local.
 - Deploy : installe le livrable dans un dépôt distant.

Construction logicielle avec Maven : Le cycle de vie Clean

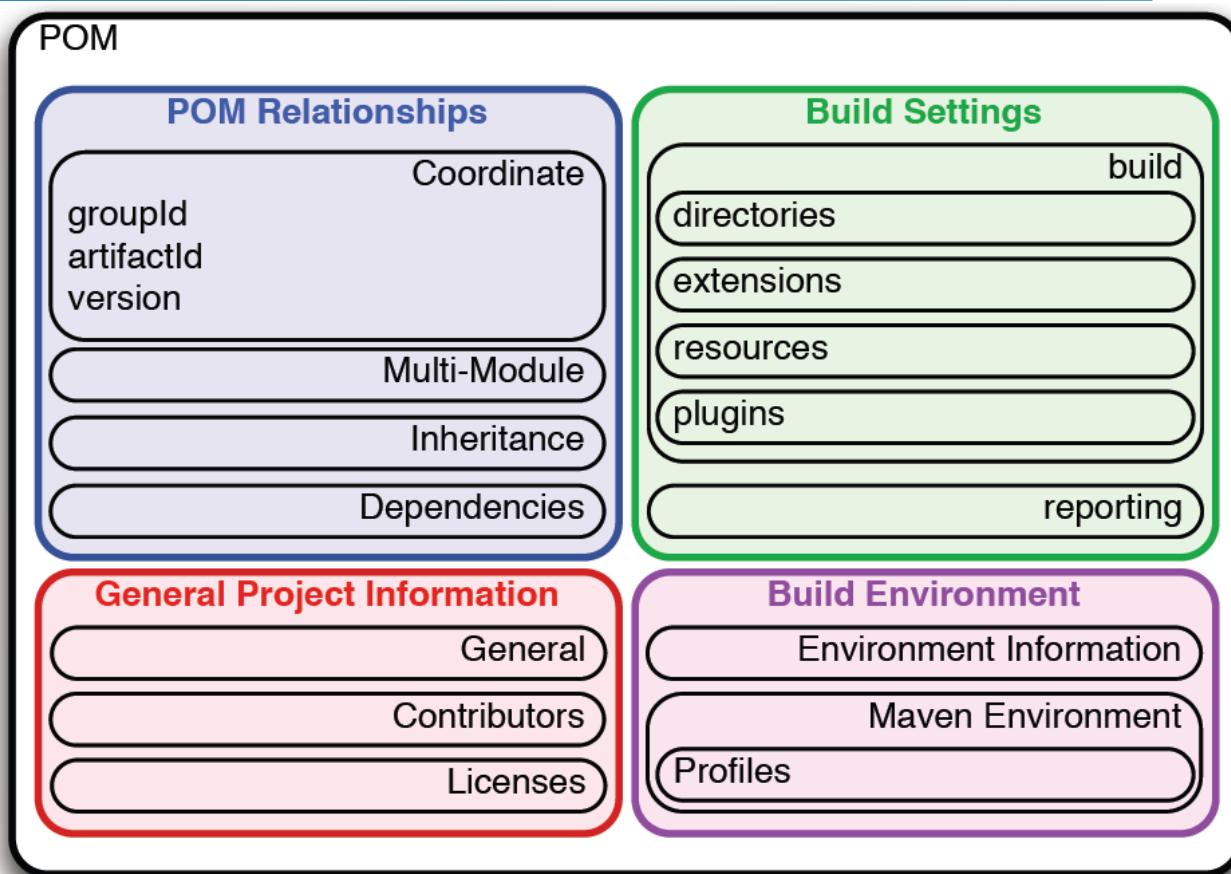
- Le cycle de vie Clean permet de nettoyer la précédente construction.
- Composé de 3 phases :
 - Pre-clean
 - Clean
 - Post-clean
- La commande mvn clean supprime le répertoire de la construction.
- Le comportement de cette commande peut être personnalisé en précisant le goal à exécuter mvn clean:goal.

Construction logicielle avec Maven : Le cycle de vie Site

- Le cycle de vie Site permet de générer la documentation du projet.
- Composé de 4 phases :
 - Pre-site
 - Site
 - Post-site
 - Site-deploy

Construction logicielle avec Maven

- POM – Structure du fichier



Construction logicielle avec Maven

- POM - XML

```
<project ...>
    <parent />
    <groupId /> <artifactId /> <version />
    <packaging />
    <developers /> <contributors />
    <scm>... </scm>

    <build>
        <plugins>....</plugins>
    </build>

    <dependencies>....</dependencies>
    <repositories>....</repositories>
    <pluginRepositories>... </pluginRepositories>
    <profiles>...</profiles>
    <reporting>... </reporting>
</project>
```

Construction logicielle avec Maven : POM - Minimaliste

```
<project>
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.Maven</groupId>
  <artifactId>MavenApp</artifactId>
  <version>1.0.0</version>
</project>
```

- Compile le code source du projet.
- Exécute les tests.
- Génère un package jar.
- Déploie le package jar généré dans un dépôt local.

Construction logicielle avec Maven : POM - Coordonnées

- **GroupId :**
 - En général, le nom du domaine à l'envers.
- **ArtifactId :**
 - Nom de l'artefact généré par le projet.
- **Version :**
 - `majorVersion.minorVersion.incrementalVersion-qualifier.`
 - Permet d'identifier le numéro du build du projet.
- **Packaging :**
 - Type de paquetage à produire (jar, war, ear, etc.).
- **Le tuple GroupId, ArtifactId, Version constitue l'identifiant de l'artefact du projet.**

Construction logicielle avec Maven : Structure des répertoires projet

- pom.xml
- src
 - main
 - java
 - resources
 - test
 - java
 - resources
- target
 - classes
 - test-classes

Construction logicielle avec Maven : Gestion des dépendances

- Un des points forts de maven qui simplifie le développement des projets Java.
- Permet une construction facile du classpath.
- Une résolution des dépendances transitive :
 - Permet de récupérer automatiquement les dépendances d'une dépendance et de les inclure dans le projet.
 - Il n'y a pas de limites sur le nombre de niveau de résolution des dépendances.
 - Possibilité d'exclure des dépendances récupérées par transitivité.

Construction logicielle avec Maven : Gestion des dépendances

- Spécifiées par :
 - Un identifiant (group/artifact/version)
 - Une portée (scope).
- Version
 - 1.2.3 : si possible 1.2.3 si non plus récent.
 - [1.2,1.3] : de 1.2 à 1.3 inclus.
 - (1.2,1.4) : à partir de 1.2, mais avant 1.4.
 - (,2.0) : antérieur à 2.0.
 - [1.4,] : à partir de 1.4.
 - [1.2.3] : 1.2.3 absolument.

Construction logicielle avec Maven : Gestion des dépendances

- Portée

- Compile : nécessaire à la compilation, et inclus dans le paquetage (portée par défaut).
- Provided : nécessaire à la compilation, non packagé (ex : Servlets).
- Runtime : nécessaire pour exécution et test, mais pas compilation (exemple : driver jdbc).
- Test : nécessaire uniquement pour les tests (exemple : junit).
- System : comme provided + chemin dans le système de fichiers à préciser.

Construction logicielle avec Maven

- Gestion des dépendances

```
<dependency>
  <groupId>org.slf4j</groupId>
  <artifactId>slf4j-api</artifactId>
  <version>1.7.7</version>
  <scope>compile</scope>
</dependency>
```

Construction logicielle avec Maven : Gestion des Repositories

- Gère les artefacts des projets ainsi que leurs dépendances.
- Peut stocker des artefacts de type jar, war, ejb, rar, ear.
- Deux types de repository existent :
 - Local.
 - Remote.
- Un repository peut être snapshot ou release.
- Repositories par défaut :
 - Local sous le home de l'utilisateur.
 - <http://repo.maven.apache.org/maven2/>

Construction logicielle avec Maven

- Gestion des Repositories

```
<repository>
  <id>jboss-repo</id>
  <name>The Release JBoss maven repo</name>
  <url>http://repository.jboss.org/maven2</url>
  <releases>
    <enabled>true</enabled>
  </releases>
</repository>
<repository>
  <id>jboss-snapshot-repo</id>
  <name>The Snapshot JBoss maven repo</name>
  <url>http://snapshots.jboss.org/maven2</url>
  <snapshots>
    <enabled>true</enabled>
  </snapshots>
</repository>
```

Questions & Réponses

Q&R

Sommaire

- **Chapitre 1 : Introduction**
- **Chapitre 2 : L'intégration Continue**
- **Chapitre 3 : Automatisation des Builds**
- **Chapitre 4 : Le gestionnaire de sources**
- **Chapitre 5 : Le serveur d'intégration continue : Jenkins**
- **Chapitre 6 : La gestion des livrables**

Le gestionnaire de sources

- Agenda
 - Les objectifs de la gestion de sources
 - Les fonctionnalités
 - Les différents gestionnaires de sources
 - Les problématiques d'intégration des changements

Les objectifs de la gestion de sources

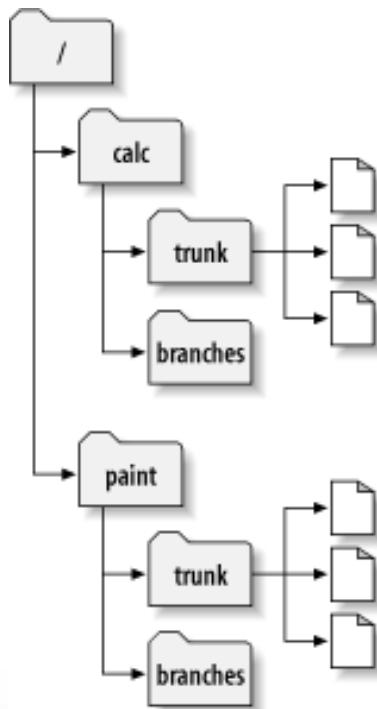
- Gérer les activités de lecture, écriture et fusion sur les sources du projet.
- Conserver un historique des modifications apportées aux sources:
 - Par qui, quand, quoi ?
- Permettre de revenir en arrière en cas de besoin.
- Permettre travailler en équipe sur les mêmes sources d'un projet.
 - Gérer les accès concurrents.
 - Gérer les conflits.
- Permettre de travailler simultanément sur plusieurs versions d'un logiciel.

Les objectifs de la gestion de sources

- Permet de retrouver un fichier source supprimé.
- Fonctionne avec tout type de fichier (.txt, .php, .java, .jpg, .exe, ...).
- Garantir la sécurité de la configuration du logiciel:
 - Autorisation.
 - Confidentialité.
 - Intégrité.
 - Disponibilité.

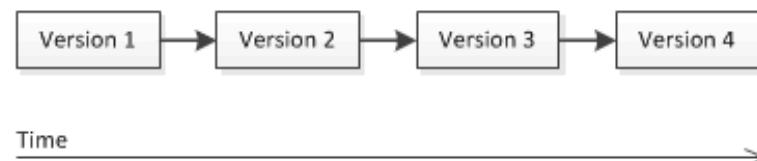
Les fonctionnalités : Repository

- Un dépôt (local ou distant) qui contient toutes les versions des sources du projet.
- Une de données des changements apportés à la configuration du logiciel.
- Le dépôt a une structure arborescente :

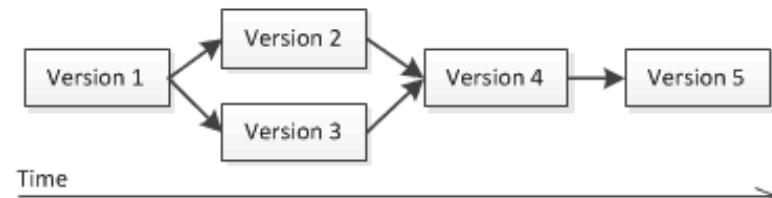


Les fonctionnalités : Version

- Une version ou révision est une instance d'un fichier source qui est strictement distincte des autres instances.
- Les versions peuvent se succéder de manière linéaire dans le temps.



- En cas de modifications concurrentes d'une version des splits et des merges peuvent avoir lieu.



Les fonctionnalités

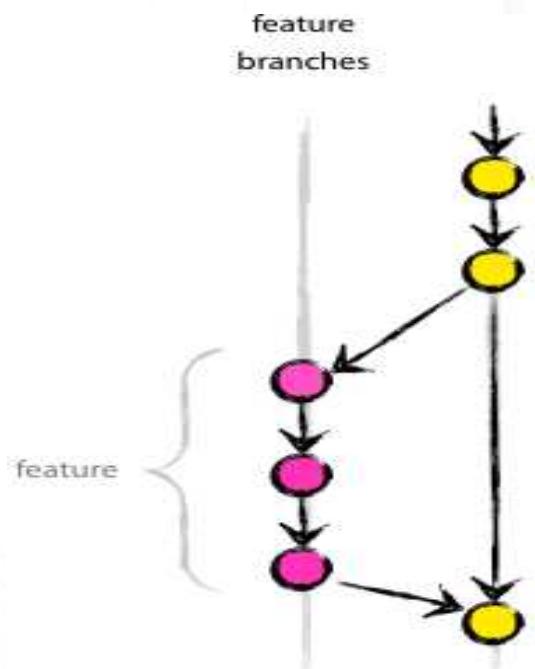
- Version (Exemple)



LE GESTIONNAIRE DE SOURCES

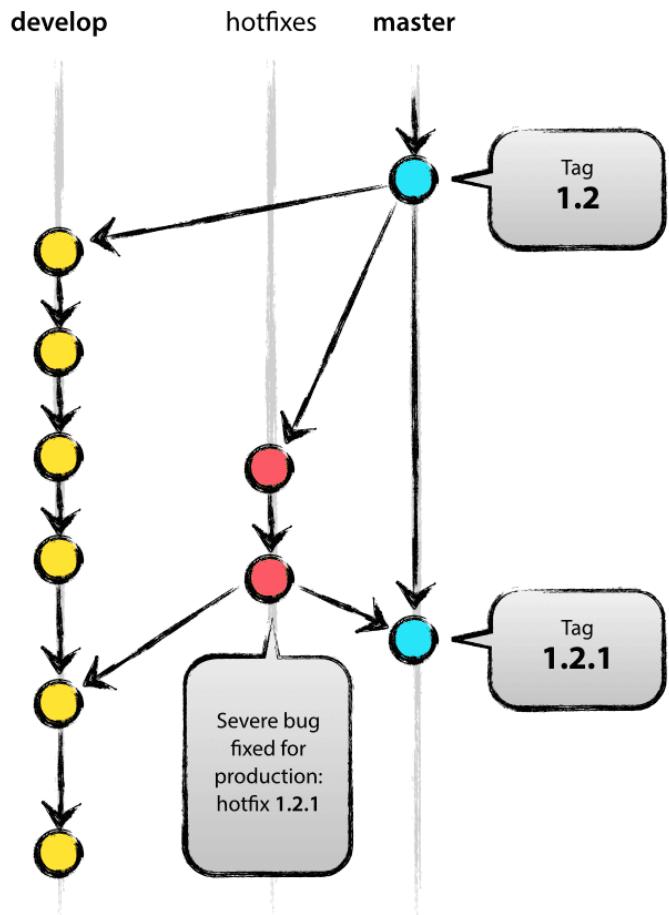
Les fonctionnalités : Branche

- Permet de développer en parallèle plusieurs versions du logiciel.
- Permet de corriger un problème sur une ancienne version du logiciel.
- Permet de fusionner après une divergence.



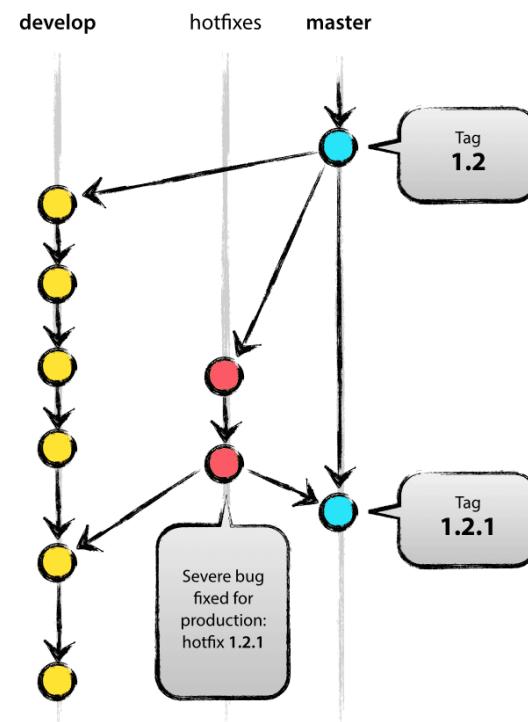
Les fonctionnalités

- Branche (Exemple)



Les fonctionnalités : Tag

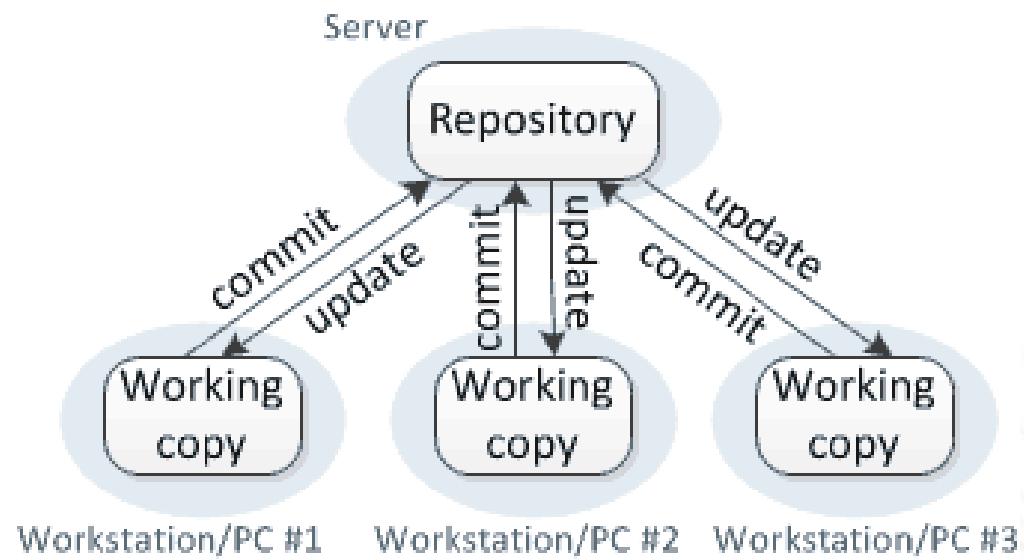
- Donner un nom explicite à une version du logiciel pour pouvoir y accéder facilement.
- Permet de définir les versions du projet.
- Permet de nommer des branches.



Les différents gestionnaires de sources : Modèle centralisé (1)

- Centralized Version Control System – CVCS

Centralized version control



Les différents gestionnaires de sources : Modèle centralisé (2)

- Le dépôt est stocké dans un endroit partagé.
- Chaque développeur a une copie de travail du dépôt.
- Utilise le mode de verrouillage pour gérer les développements concurrents sur un même fichier source.
- Il est nécessaire d'avoir la connexion au dépôt pour commiter ses modifications ou mettre à jour sa copie de travail.

Les différents gestionnaires de sources : Modèle centralisé (3)

- Avantages :
 - Technologie éprouvée.
 - Structure simple.
 - Gestion et utilisation simples à mettre en œuvre.
 - Largement disponible (IDE, Forges).
- Faiblesses :
 - Très sensible aux pannes, impossible de travailler hors connexion.
 - Inadapté aux très grands projets, le temps de mise à jour du dépôt est trop long.

Les différents gestionnaires de sources : Modèle centralisé – Exemple Subversion

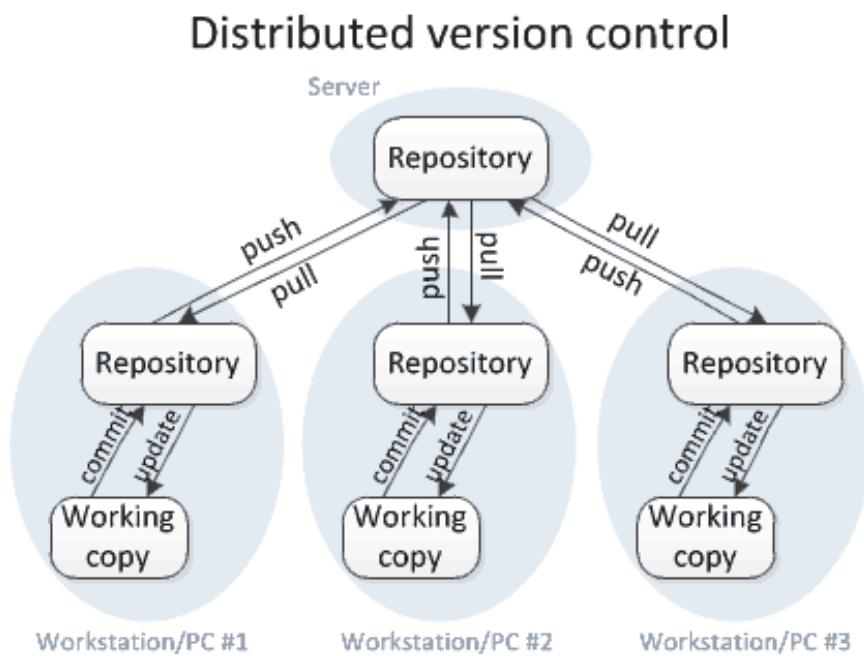
- Crée en 2000 pour remplacer CVS.
- Logiciel libre.
- Multi-plateforme (linux, windows, MacOS, ...).
- Documentations très riches, forums actifs.
- Implémente des protocoles réseaux sécurisés (HTTPS) .
- Interfaces graphiques.
- Intégré dans plusieurs IDE tel que Eclipse.



LE GESTIONNAIRE DE SOURCES

Les différents gestionnaires de sources : Modèle distribué (1)

- Distributed Version Control System – DVCS



Les différents gestionnaires de sources : Modèle distribué (2)

- Chaque développeur a sa copie avec des branches et des tags privés.
- Utilise le mode de fusion pour gérer les développements concurrents sur un même fichier source.

Les différents gestionnaires de sources : Modèle distribué (3)

- Avantages :

- Travail déconnecté, accès aux dépôts en local sur le poste de travail du développeur.
- Moins sensible aux pannes.
- Rapidité, le réseau n'est utilisé que pour les opérations de synchronisation.
- Branches privées.
- Modèle de développement autorisé très souple.

- Faiblesses :

- Gestion et utilisation plus compliquées.
- Risque de divergence, peut devenir très complexe structurellement.

Les différents gestionnaires de sources : Modèle distribué – Exemple Git

- Crée en 2005 par Linus Torvalds pour la gestion des sources du noyau Linux.
- Logiciel libre.
- Multi-plateforme (linux, windows, MacOS, ...).
- Documentations très riches, forums actifs.
- Implémente des protocoles réseaux sécurisés (HTTPS) .
- Interfaces graphiques.
- Intégré dans plusieurs IDE tel que Eclipse.
- Ne stocke que le delta entre deux versions d'un objet.



Exercice

Utilisation de Git

LAB

LE GESTIONNAIRE DE SOURCES



© Brain Consulting – Tous droit de reproduction ou de diffusion réservés

23/08/2018

106

Problématiques de fusion des changements

- Plus la quantité de code publiée est grosse plus le risque de conflit augmente.
- La publication régulière du code source :
 - Réduit le risque de conflit.
 - Facilite l'identification de la modification à l'origine du problème.
- Pour pouvoir publier régulièrement le code source il faut :
 - Publier les modifications apportées à la fin de chaque tâche de développement.
 - Faire des petits changements et ne pas apporter des changements à plusieurs composants dans une seule tâche.

Problématiques de fusion des changements : Bonnes pratiques

- Essayez de travailler le plus possible avec des fichiers texte et le moins possible avec des fichiers binaires.
- Faites des update le plus souvent et régulièrement possible, et obligatoirement avant de commencer à travailler sur un fichier.
- Faites des commit réguliers, dès que vous avez fini de travailler sur un fichier, et si la tâche prend du temps, un commit intermédiaire permettra de sauvegarde votre travail.
- Puez un planning pour savoir qui peut travailler et quand sur les fichiers binaires notamment.

Questions & Réponses

Q&R

LE GESTIONNAIRE DE SOURCES

Sommaire

- **Chapitre 1 : Introduction**
- **Chapitre 2 : L'intégration Continue**
- **Chapitre 3 : Automatisation des Builds**
- **Chapitre 4 : Le gestionnaire de sources**
- **Chapitre 5 : Le serveur d'intégration continue : Jenkins**
- **Chapitre 6 : La gestion des livrables**

Le serveur d'intégration continue Jenkins

- Agenda

- Le serveur Jenkins
- Le rôle du serveur Jenkins
- Les grandes fonctionnalités
- Configuration du serveur Jenkins
- La gestion des plugins Jenkins
- Les plugins les plus utilisés
- Gestion des utilisateurs et des autorisations
- Gestion de l'espace disque
- Le monitoring du serveur Jenkins
- Création de Job Jenkins

Le serveur Jenkins



Jenkins

Le serveur d'intégration continue jenkins

23/08/2018

112

Le serveur Jenkins

- Jenkins a été créé par Kohsuke KAWAGUCHI en 2004 sous le nom Hudson.
- Jenkins est un fork du serveur Hudson depuis 2011.
- Jenkins est développé en Java et Groovy.
- Jenkins s'exécute dans un conteneur de Servlet de type Apache Tomcat.
- Jenkins est perçu comme le serveur d'intégration continue le plus populaire du marché.
- Jenkins est un logiciel libre, publié sous licence MIT.
- Jenkins est supporté par une grande communauté et une documentation complète.
- Le site de référence de Jenkins est :
 - <https://jenkins.io/>

Le rôle du serveur Jenkins

- Le serveur d'intégration continue est l'orchestrateur de l'usine logicielle.
- Il permet de faire collaborer plusieurs outils contribuant à l'exécution du processus d'intégration continue.
- Jenkins est un outil open source simple, extensible et facile à utiliser qui fournit des services d'intégration continue pour le développement des applications.
- Jenkins est un outil de construction continue qui permet aux équipes de se concentrer sur leur travail pour créer de la valeur métier en automatisant les processus de construction, de gestion des artefacts et de déploiement.

Le rôle du serveur Jenkins

- Jenkins peut être utilisé par des équipes de toutes tailles, pour des projets dans une grande variété de langages de programmation, y compris Java, .NET, Ruby, Groovy, Grails, PHP etc.
- L'interface utilisateur de Jenkins est simple, intuitive.

Les grandes fonctionnalités

- Jenkins est l'un des serveurs d'intégration continue les plus populaires sur le marché en raison de :
 - Installation facile sur différents systèmes d'exploitation.
 - Mise à jour facile, Jenkins a des cycles de livraison très rapides.
 - Facilement configurable via une interface graphique d'administration.
 - Facilement extensible avec l'utilisation de plugins tiers, plus de 1400 plugins maintenus par une communauté hyperactive permettant la construction, le déploiement et l'automatisation de tout type de projet.
 - Interface utilisateur simple et facile à utiliser.

Les grandes fonctionnalités

- Infiniment scalable grâce à l'architecture maître-esclave qui prend en charge la génération distribuée de build pour réduire les charges sur le serveur d'intégration continue.
- Jenkins prend en charge le build de projets de type Apache Maven, Apache Ant, Gradle, Freestyle, etc.
- Jenkins prend en charge la planification de builds basée sur les expressions cron ou déclenchée par un commit dans un système de contrôle de version.
- Jenkins permet la supervision de l'exécutions de tâches, telle que la construction d'un projet.
- Jenkins prend en charge les outils de gestion de code source tels que Git, Subversion, CVS, StarTeam , ClearCase, AccuRev, Team Foundation Server, etc.

Les grandes fonctionnalités

- Jenkins permet de lancer l'exécution de commandes Shell linux et Windows lors des étapes de pré-build et de post-build.
- Jenkins supporte l'exécution des tests et l'analyse de la qualité du code source. Les résultats sont disponibles sous forme de rapport et de graphique.
- Jenkins prend en charge la publication des artefacts projets dans différents gestionnaires de dépôts d'artefact tel que Apache Archiva, Artifactory, Sonatype Nexus, etc.
- Jenkins supporte le déploiement directement dans les environnements de test ou de production.
- Jenkins supporte l'envoi de notifications liées à l'état du build.

Les grandes fonctionnalités

- Jenkins supporte la gestion des permissions avec la possibilité de raccordement à un Ldap d'entreprise.
- Jenkins supporte la surveillance de la santé du serveur.

Exercice

Installation d'un serveur Jenkins

LAB

Le serveur d'intégration continue jenkins

23/08/2018

120

Configuration du serveur Jenkins

- La configuration du serveur Jenkins est accessible aux administrateurs du serveur.
- La vue « Administrer Jenkins » permet d'accéder à :
 - La configuration du système.
 - La configuration globale des outils.
 - La configuration globale de la sécurité du serveur.
 - La gestion des plugins.
 - Les informations, les logs et les statistiques d'utilisation du système.
- La vue « Administrer Jenkins » est le point central des tâches de configuration du serveur Jenkins.

Configuration du serveur Jenkins

- La vue «Administre Jenkins » est accessible via le menu « Administre Jenkins ».



Le serveur d'intégration continue jenkins

23/08/2018

122

Configuration du serveur Jenkins : Configuration système

- La vue « Configurer le système », permet de définir des paramètres globaux du serveur Jenkins :
 - Le répertoire HOME: fournie le chemin vers où le serveur Jenkins stocke ses fichiers de configuration, exécute les construction des projets et archive les résultats
 - Le nombre d'exécuteurs: permet de configurer le nombre de tâches de construction simultanées que peut lancer le serveur Jenkins.
 - Variables d'environnement: permet de configurer des paires de clé-valeur s'appliquent à chaque construction de projet.
 - Emplacement des outils: permet de spécifier le chemin où se trouvent certains outils.

Configuration du serveur Jenkins : Configuration système

- Stratégie de nommage: permet de définir un pattern qui doit être vérifié par les noms de jobs.
- Configuration des moniteurs administratifs: permet de configurer les alertes remontées aux administrateurs à propos de l'état de l'instance Jenkins.
- Adresse email de l'administrateur système: permet de configurer l'adresse d'envoi des e-mail de notification.
- Shell: permet de configurer le chemin de l'exécutable du Shell que doit utiliser Jenkins.
- Cette vue peut être enrichie avec l'installation des plugins pour accomplir des tâches spécifiques.

Configuration du serveur Jenkins : Configuration système

- La vue « Configurer le système » est accessible via « Administrer Jenkins > Configurer le système ».



Le serveur d'intégration continue jenkins

23/08/2018

125

Configuration du serveur Jenkins

- Configuration système



The screenshot shows the Jenkins 'Configure System' interface. The URL in the browser is 127.0.0.1:7070/configure. The page has a sidebar with links like 'Nouveau Item', 'Utilisateurs', 'Historique des constructions', 'Administre Jenkins', 'Mes vues', and 'New View'. The main content area includes sections for 'File d'attente des constructions' (with 'File d'attente des constructions vide'), 'État du lanceur de compilations' (with '1 Au repos' and '2 Au repos'), and 'Nb d'exécuteurs' (set to 2). There are also sections for 'Libellés', 'Utilisation', 'Période d'attente', 'Nombre de tentatives de checkout SCM', and 'Restreindre le nommage de projet'. Other sections include 'Propriétés globales', 'Emplacement des outils', 'Variables d'environnement', 'Usage Statistics' (with a checkbox for sending anonymous usage statistics), and 'Administrative monitors configuration'. At the bottom, there are fields for 'Jenkins Location' (URL: http://127.0.0.1:7070/), 'Adresse email de l'administrateur système' (Address pas encore configurée <nobody@nowhere>), and 'Shell' (Executable pour le Shell). Buttons at the bottom are 'Enregistrer' and 'Appliquer'.

Le serveur d'intégration continue jenkins

Configuration du serveur Jenkins : Configuration globale des outils

- La vue « Configuration globale des outils » fournie des détails relatifs à tous les outils disponibles sur le serveur et qui peuvent être utilisés pour effectuer certaines tâches dans le cadre de la construction des projets.
- Dans cette vue il est possible de configurer plusieurs versions d'un même outil.
 - Par exemple, nous pouvons configurer Java 7, Java 8 et Java 9 pour répondre à des besoins différents de construction de projets.

Configuration du serveur Jenkins : Configuration globale des outils

- La vue par défaut « Configuration globale des outils » permet de configurer les installations des JDK et Maven qui peuvent être utilisées par Jenkins pour construire des projets.
- Cette vue peut être enrichie avec l'installation des plugins pour accomplir des tâches spécifiques.

Configuration du serveur Jenkins : Configuration globale des outils

- La vue «Configuration globale des outils» est accessible via « Administrer Jenkins> Configuration globale des outils».



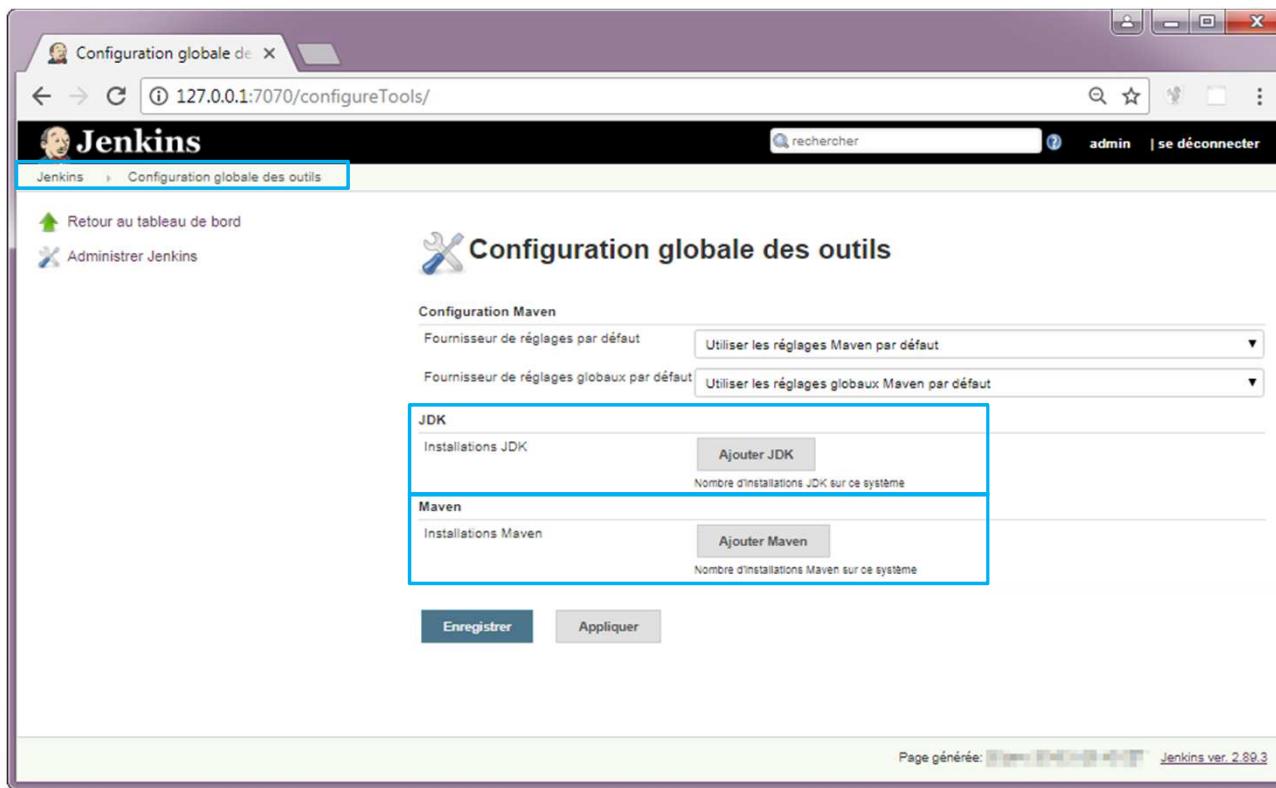
Le serveur d'intégration continue jenkins

23/08/2018

129

Configuration du serveur Jenkins

- Configuration globale des outils



130

Configuration du serveur Jenkins : Configuration globale de la sécurité

- La vue « Configurer la sécurité globale » permet de voir et configurer les paramètres de sécurité disponibles dans le serveur Jenkins:
 - Activer / désactiver la sécurité.
 - Configurer la gestion d'accès et les autorisations.
 - Etc.
- Jenkins prend en charge plusieurs modèles de sécurité et peut s'intégrer à différents référentiels d'utilisateurs.

Configuration du serveur Jenkins : Configuration globale de la sécurité

- La vue « Configurer la sécurité globale » est accessible via « Administrer Jenkins > Configurer la sécurité globale ».



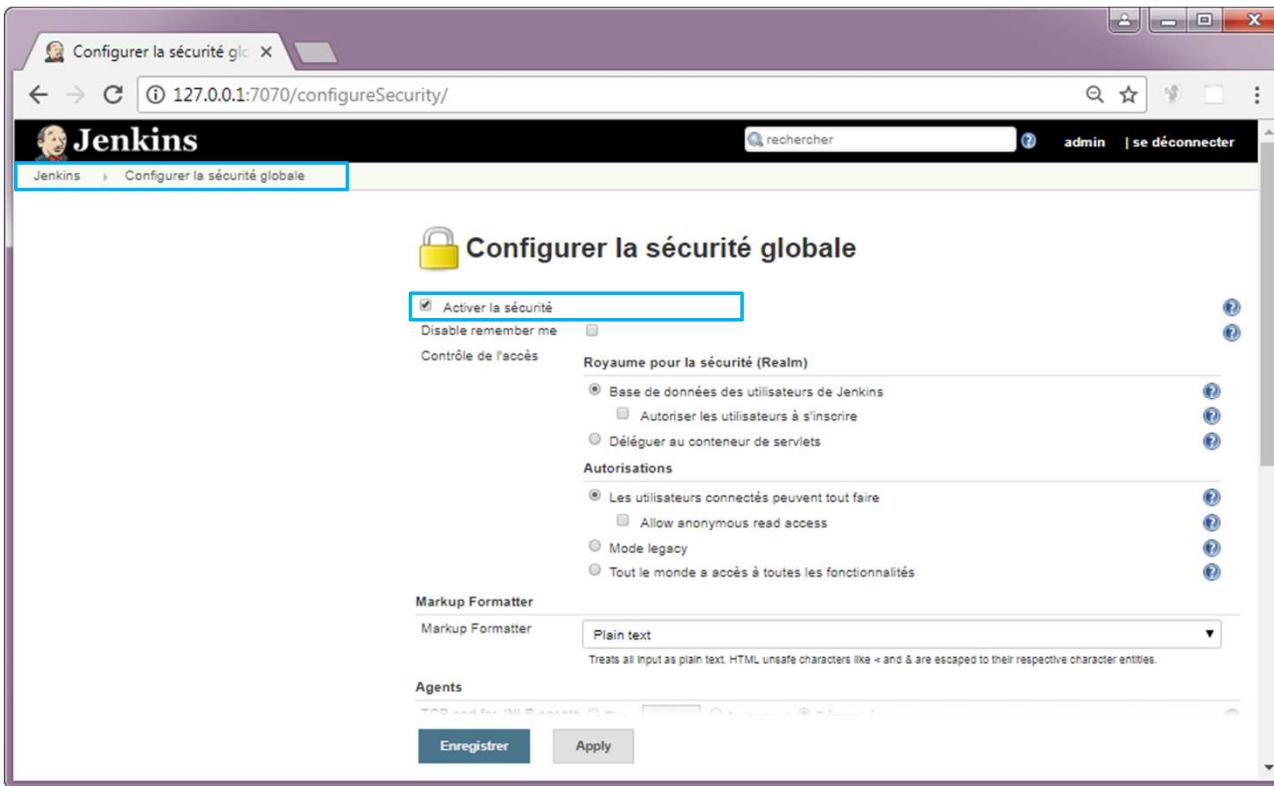
Le serveur d'intégration continue jenkins

23/08/2018

132

Configuration du serveur Jenkins

- Configuration globale de la sécurité



Exercice

Configuration du JDK et Maven du serveur Jenkins

LAB

Le serveur d'intégration continue jenkins

23/08/2018

134

La gestion des plugins Jenkins

- Les plugins constituent le principal moyen pour enrichir les fonctionnalités du serveur d'intégration continue Jenkins.
- Les plugins Jenkins permettent de répondre aux besoins spécifiques d'une organisation, d'un projet ou d'un l'utilisateur.
- Il existe plus de 1400 plugins qui peuvent être installés sur Jenkins.
- Il existe différentes catégories de plugins disponibles comme les plugins de :
 - Intégration des différents outils de gestion de code source.
 - Intégration des différents outils de construction.
 - Intégration des différents frameworks de test.

La gestion des plugins Jenkins

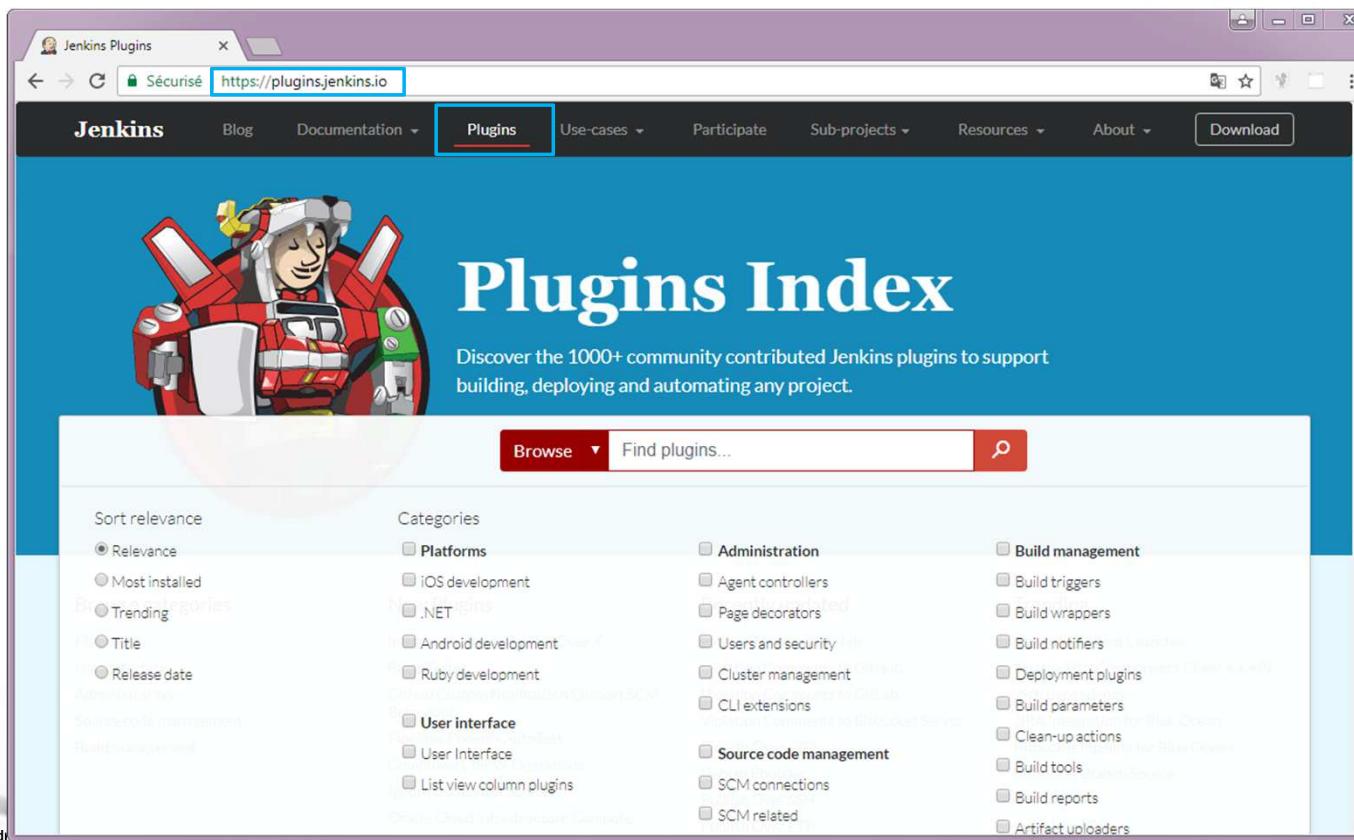
- Intégration des différents frameworks et outils d'analyse de la qualité du code source.
- Intégration des différents outils de gestion des artefacts.
- Authentification et gestion des utilisateurs.
- Administration du serveur Jenkins.
- Les plugins peuvent être téléchargés automatiquement, avec leurs dépendances, à partir du Centre de mise à jour Jenkins.
- Le centre de mise à jour est un service exploité par le projet Jenkins qui fournit un inventaire des plugins open source qui ont été développés et maintenus par divers membres de la communauté Jenkins.

La gestion des plugins Jenkins

- Un plugin installé dans Jenkins se présente comme un paramètre ou un élément configurable dans :
 - La configuration du système Jenkins.
 - La configuration d'un job Jenkins.

La gestion des plugins Jenkins

- La liste de tous les plugins Jenkins est disponible sur le site <https://plugins.jenkins.io>



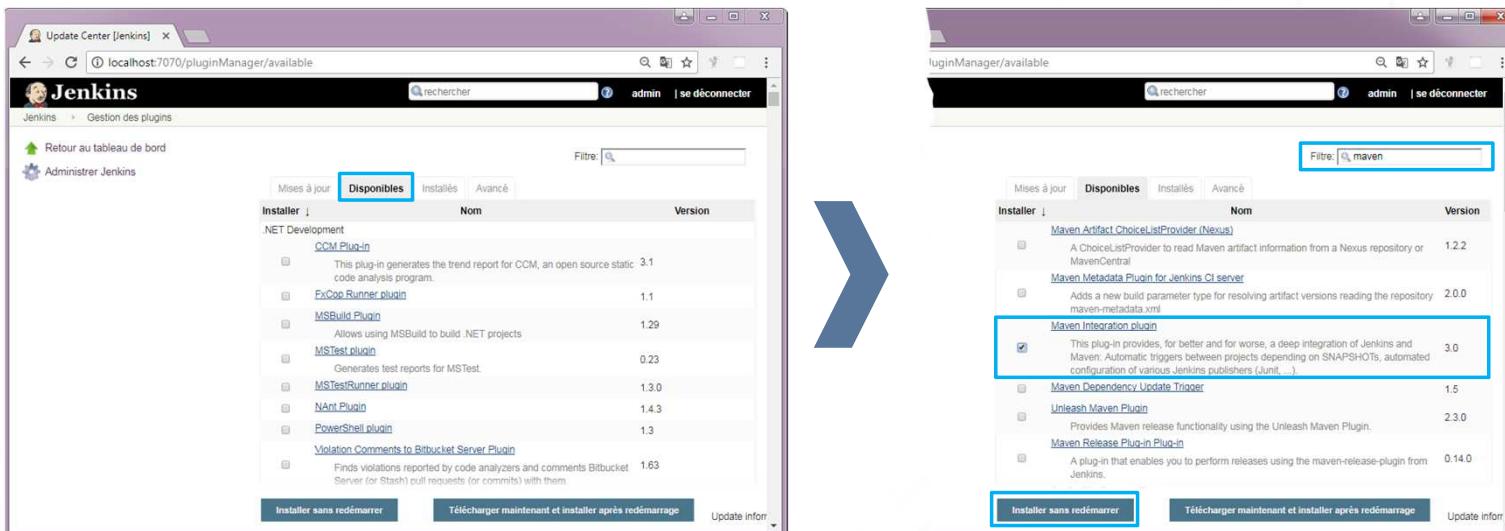
La gestion des plugins Jenkins

- La manière la plus simple pour gérer les plugins Jenkins est accessible via la vue « Administrer Jenkins > Gestion des plugins », disponible pour les administrateurs de Jenkins.
 - La gestion des plugins est également disponible via la l'interface de ligne de commande Jenkins CLI.



La gestion des plugins Jenkins

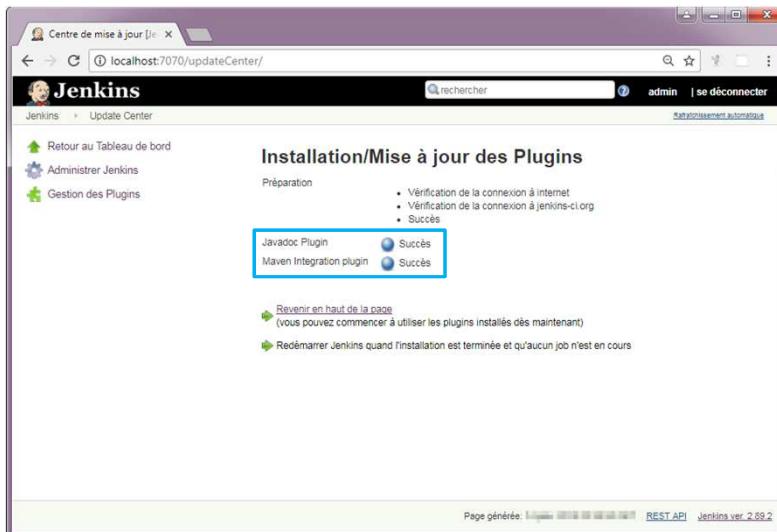
- Cliquez sur l'onglet disponible et choisissez parmi la liste de plugins disponibles les plugins à installer.
- Cliquez sur le bouton Installer sans redémarrer en bas de la page. Ceci permettra de télécharger les plugins sélectionnés et les installer dans Jenkins.



Le serveur d'intégration continue jenkins

La gestion des plugins Jenkins

- L'installation des plugins Jenkins est transitive. L'installation d'un plugin implique l'installation de toutes ses dépendances.



23/08/2018

141

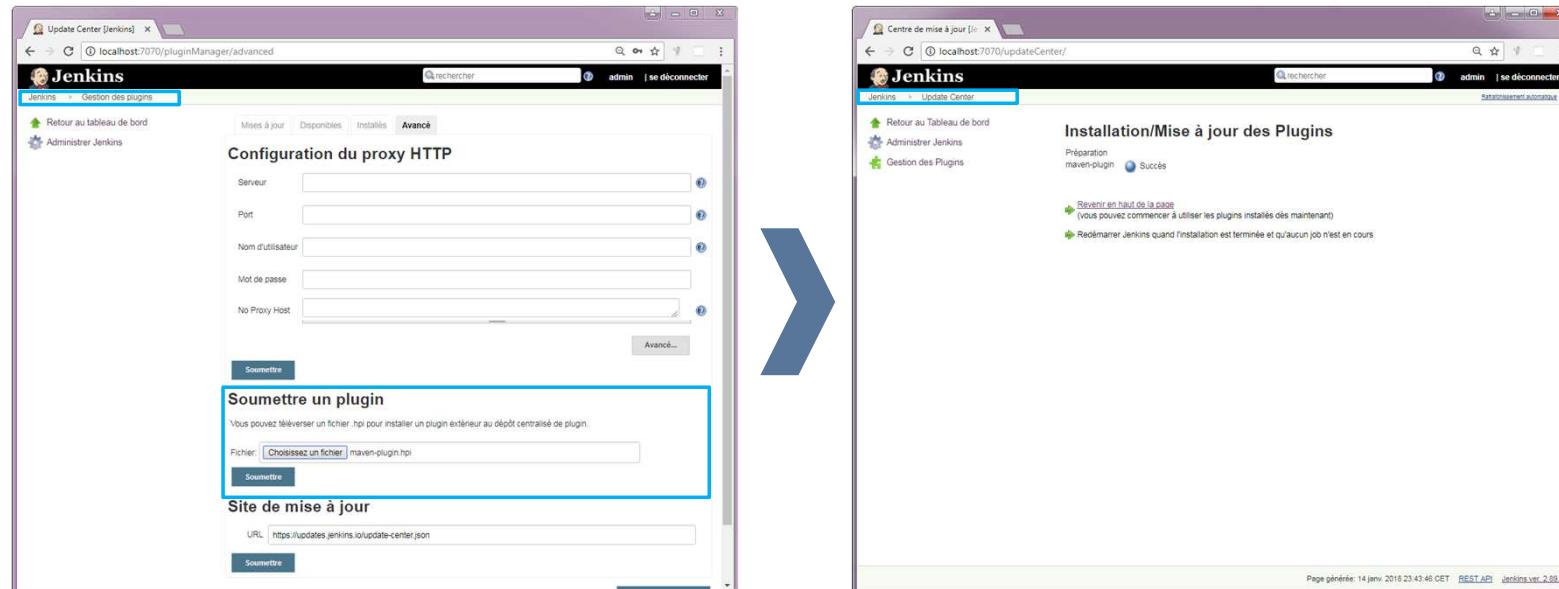
- Installation avancée d'un plugin

La gestion des plugins Jenkins

- Il peut être nécessaire d'installer une ancienne version ou une version hors-ligne d'un plugin, dans ce cas, vous pouvez télécharger le plugin à partir du site <https://updates.jenkins-ci.org/download/plugins>
- Vous pouvez ensuite utiliser l'option Télécharger pour télécharger le plugin manuellement dans Jenkins.
- Cette option est accessible via la vue « Administrer Jenkins > Gestion des plugins », disponible pour les administrateurs de Jenkins sous l'onglet Avancé.

La gestion des plugins Jenkins

- Installation avancée d'un plugin

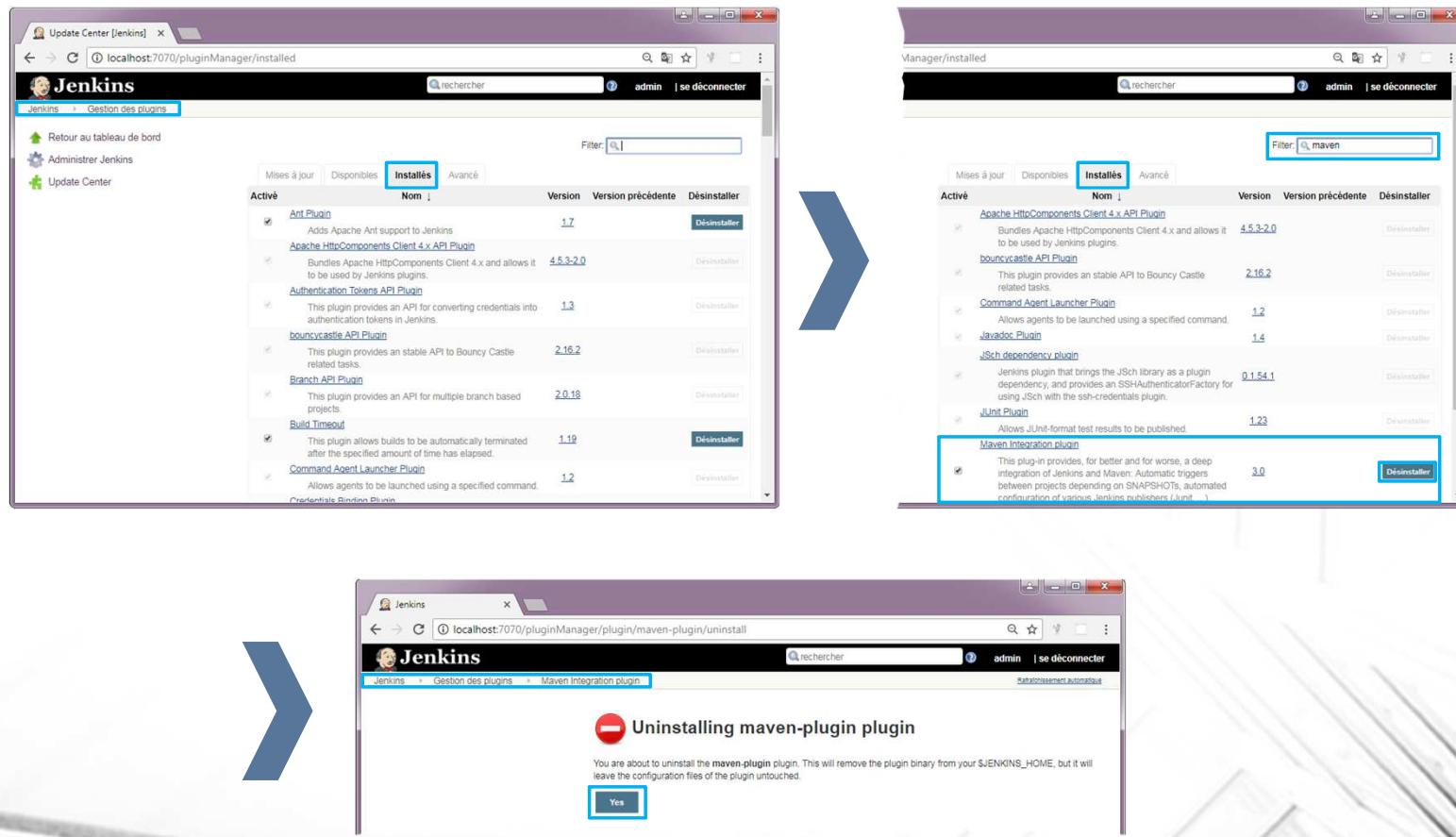


Le serveur d'intégration continue jenkins

La gestion des plugins Jenkins : Désinstaller un plugin

- La désinstallation des plugins est accessible via la vue « Administrer Jenkins > Gestion des plugins », disponible pour les administrateurs de Jenkins.
- Cliquez sur l'onglet Installé. Certains plugins auront l'option de désinstallation. Vous pouvez cliquer sur ces boutons pour désinstaller les plugins.
- Assurez-vous de redémarrer votre instance Jenkins après la désinstallation.

La gestion des plugins Jenkins : Désinstaller un plugin

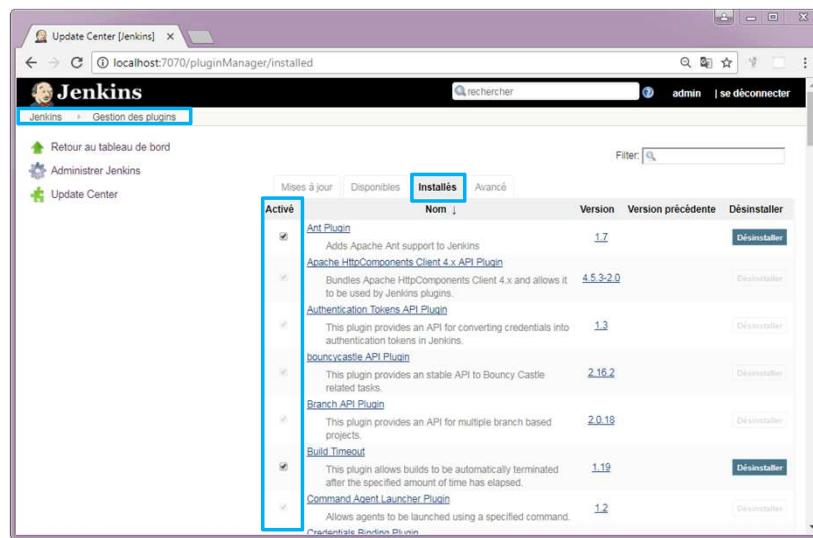


La gestion des plugins Jenkins : Désactiver un plugin

- Il possible de désactiver un plugin sans avoir à le désinstaller.
- Jenkins continuera à reconnaître que le plugin est installé, mais il ne démarrera pas le plugin, et aucune extension apportée par ce plugin ne sera visible.
- La désactivation des plugins est accessible via la vue « Administrer Jenkins> Gestion des plugins », disponible pour les administrateurs de Jenkins
- Sous l'onglet Installé, il faut décocher l'option activé du plugin à désactiver.

La gestion des plugins Jenkins

- Désactiver un plugin



Le serveur d'intégration continue jenkins

23/08/2018

147

Les plugins les plus utilisés

- Le plugin Git: permet à Jenkins d'utiliser GIT comme serveur de gestion de code source.
- Le plugin GitLab: Ce plugin est un déclencheur de construction qui permet à GitLab de déclencher des constructions Jenkins lorsque le code est publié ou qu'une requête de fusion est créée.
- Le plugin Maven integration: permet à Jenkins d'intégrer Apache Maven comme outil de construction des projets Java.
- Le plugin Maven Release: permet l'exécution des cibles Maven release:prepare et release:perform.
- Le plugin Workspace Cleanup: ce plugin supprime l'espace de travail avant ou après la terminaison de la construction.

Les plugins les plus utilisés

- Le plugin Global Build Stats: permet de connaitre combien de constructions de projets se produisent quotidiennement et hebdomadairement. Combien de temps est pris par les différentes constructions, quelle est la période d'attente. Ce plugin fournit assez de données pour répondre à ces questions.
- Le plugin Console Column: permet un accès plus rapide à la dernière sortie console de l'exécution d'un job Jenkins.
- Le plugin Green Balls: ce plugin permet de modifier Jenkins pour utiliser des balles vertes au lieu de bleu pour les constructions réussies.
- Le plugin Javadoc: permet d'afficher la javadoc dans Jenkins.

Les plugins les plus utilisés

- Le plugin **Folders**: permet de créer des dossiers pour organiser les jobs. Les utilisateurs peuvent définir des taxonomies personnalisées (par exemple, par type de projet, type d'organisation).
- Le plugin **JobConfigHistory**: Ce plugin sauvegarde une copie du fichier de configuration d'un job suite à chaque modification effectuée sur le job.

Exercice

Installation des plugins Git et Maven Integration

LAB

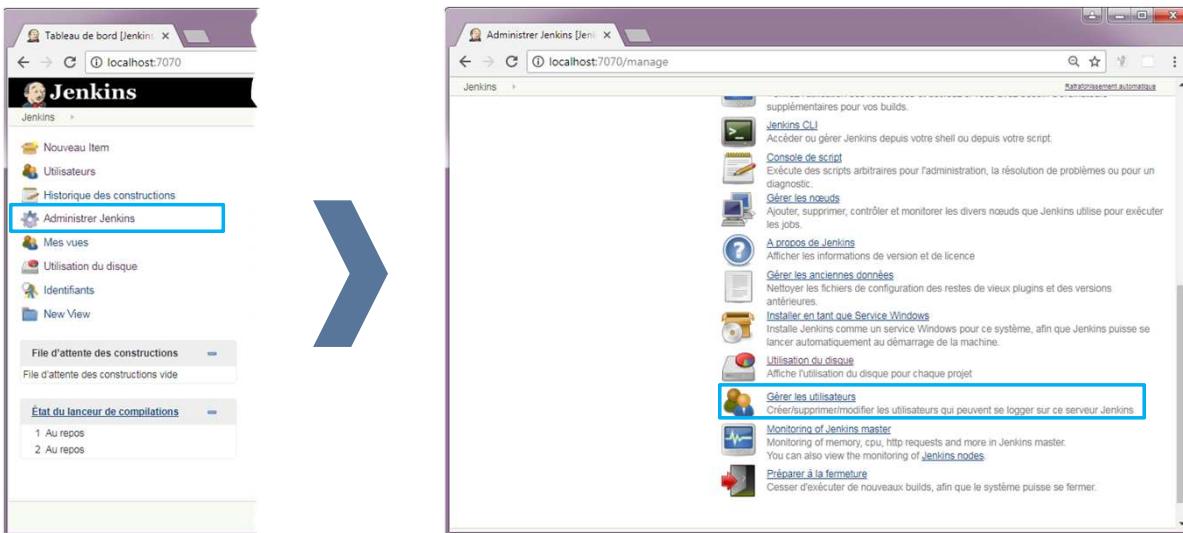
Le serveur d'intégration continue jenkins

Gestion des utilisateurs et des autorisations : Gestion des utilisateurs

- Jenkins prend en charge plusieurs modèles de sécurité et peut s'intégrer à différents référentiels d'utilisateurs.
- Par défaut, les utilisateurs sont stockés dans la base de données des utilisateurs de Jenkins paramétrée dans la vue «Administre Jenkins > Configurer la sécurité globale».
- La gestion des utilisateurs est accessible aux administrateurs de Jenkins via la vue «Administre Jenkins > Gérer les utilisateurs».

Gestion des utilisateurs et des autorisations

- La gestion des utilisateurs est accessible via la vue «Administre Jenkins > Gérer les utilisateurs».



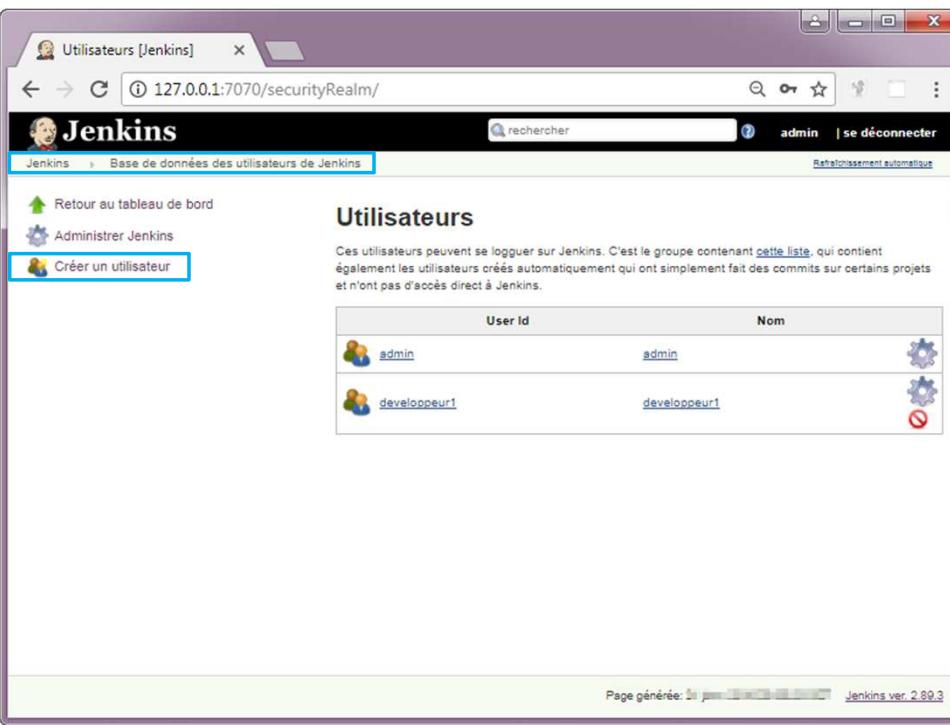
Le serveur d'intégration continue jenkins

23/08/2018

153

Gestion des utilisateurs et des autorisations

- La vue « Gérer les utilisateurs » permet de créer/supprimer/modifier les utilisateurs qui peuvent se logger sur le serveur Jenkins.



A screenshot of a web browser displaying the Jenkins 'Utilisateurs [Jenkins]' page. The URL is 127.0.0.1:7070/securityRealm/. The page title is 'Jenkins - Base de données des utilisateurs de Jenkins'. On the left, there's a sidebar with links: 'Retour au tableau de bord', 'Administrer Jenkins', and 'Créer un utilisateur' (which is highlighted with a blue border). The main content area is titled 'Utilisateurs' and contains a table with two rows:

| User Id | Nom | |
|--------------|--------------|--|
| admin | admin | |
| developpeur1 | developpeur1 | |

At the bottom of the page, it says 'Page générée: 23/08/2018 Jenkins ver. 2.89.3'.

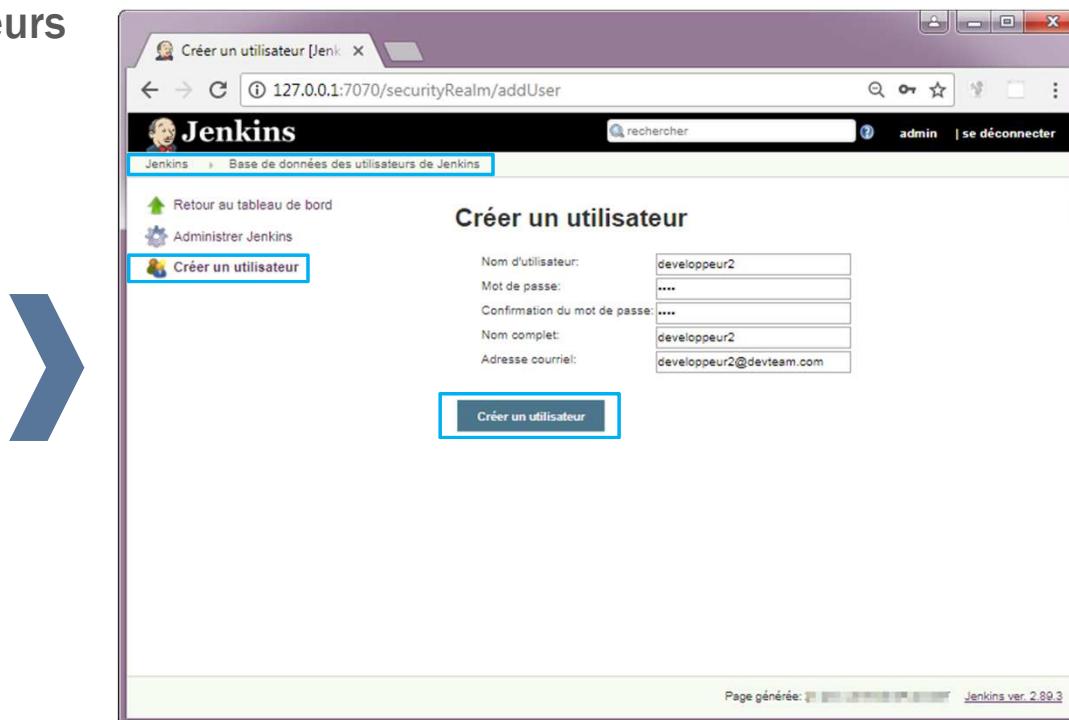
Le serveur d'intégration continue jenkins

23/08/2018

154

Gestion des utilisateurs et des autorisations

- Gestion des utilisateurs



Le serveur d'intégration continue jenkins

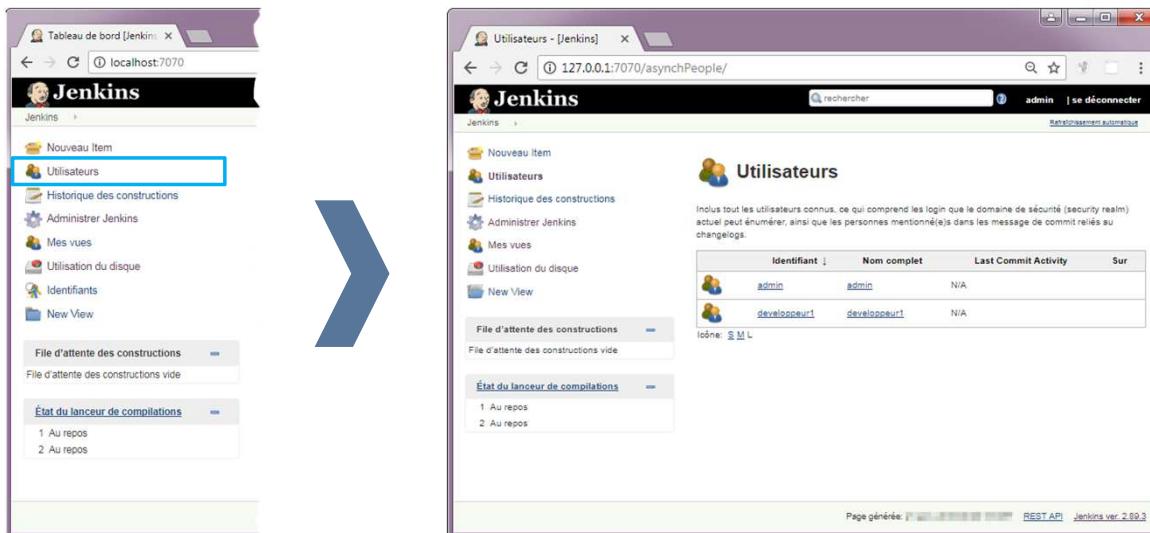
23/08/2018

155

- Gestion des utilisateurs

Gestion des utilisateurs et des autorisations

- La liste de tous les utilisateurs est également disponible via la vue « Utilisateurs ».



The image shows two screenshots of the Jenkins web interface. On the left, the main Jenkins dashboard is displayed with various links like 'Nouveau Item', 'Utilisateurs', and 'Historique des constructions'. A blue arrow points from the 'Utilisateurs' link to the right screenshot. The right screenshot shows the 'Utilisateurs' page itself, which lists two users: 'admin' and 'developpeur1'. The table has columns for 'Identifiant', 'Nom complet', 'Last Commit Activity', and 'Sur'.

| Identifiant | Nom complet | Last Commit Activity | Sur |
|--------------|--------------|----------------------|-----|
| admin | admin | N/A | |
| developpeur1 | developpeur1 | N/A | |

Le serveur d'intégration continue jenkins

23/08/2018

156

Gestion des utilisateurs et des autorisations : Gestion des autorisations

- Les autorisations peuvent être gérées en implémentant une stratégie de sécurité basée sur une matrice.
- La vue « Administrer Jenkins > Configurer la sécurité globale » permet de configurer la matrice des autorisations:
 - Il est possible d'ajoutez des groupes ou des utilisateurs et configurez la sécurité en fonction de différentes sections telles que Job, Historique des builds, Metrics, Vues, etc.

Gestion des utilisateurs et des autorisations : Gestion des autorisations

- La vue « Configurer la sécurité globale » est accessible via « Administrer Jenkins > Configurer la sécurité globale ».



Le serveur d'intégration continue jenkins

23/08/2018

158

Gestion des utilisateurs et des autorisations : Gestion des autorisations

- Chaque ligne de la matrice représente un utilisateur ou un groupe d'utilisateurs.
- Chaque colonne de la matrice représente une autorisation.

The screenshot shows a user authorization matrix interface. At the top, there is a navigation bar with links like "Accueil", "Aide", "Connexion", and "Déconnexion". Below the navigation bar, a title bar says "Gestion des utilisateurs et des autorisations : Gestion des autorisations". A large blue arrow points from the left towards the interface. The main area is titled "Autorisations" and contains a list of authorization strategies:

- Les utilisateurs connectés peuvent tout faire
- Mode legacy
- Stratégie d'autorisation matricielle basée sur les projets
- Sécurité basée sur une matrice**

The "Sécurité basée sur une matrice" option is selected and highlighted with a blue border. The interface then displays a grid where users are assigned permissions across various functional areas. The columns represent different functions: Global, Agent, Job, and Historique des builds. The rows list users: Anonymous Users, Authenticated Users, admin, and developpeur1. The "developpeur1" row is also highlighted with a blue border. The "Enregistrer" button at the bottom left is also highlighted with a blue border.

Exercice

Gestion des utilisateurs et des autorisations

LAB

Le serveur d'intégration continue jenkins

23/08/2018

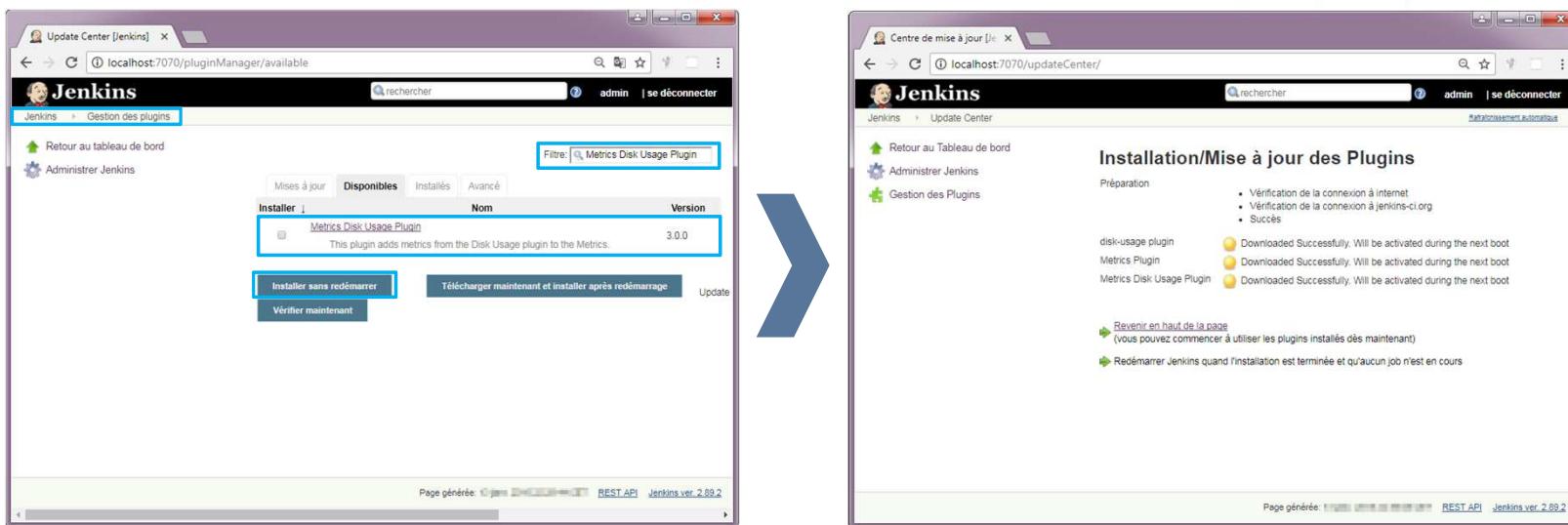
160

Gestion de l'espace disque

- Le plugin Metrics Disk Usage enregistre et rapporte l'espace disque utilisé par les projets gérés par le serveur Jenkins.
- Il permet d'isoler et de nettoyer les projets qui utilisent excessivement de l'espace disque.

Gestion de l'espace disque

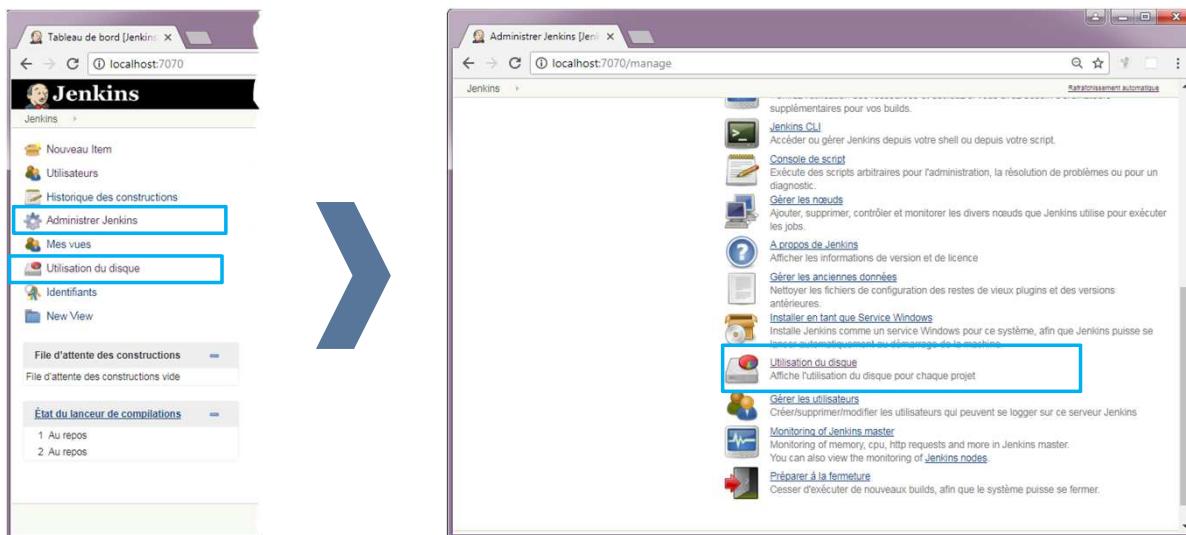
- L'installation du plugin Metrics Disk Usage est disponible via le gestionnaire de plugin dans la vue « Administrer Jenkins > Gestion des plugins » dans l'onglet Disponible.



Le serveur d'intégration continue jenkins

Gestion de l'espace disque

- La vue de « Utilisation du disque » est accessible via « Administrer Jenkins > Utilisation du disque » ou via le menu « Utilisation du disque » après installation du plugin Metrics Disk Usage.

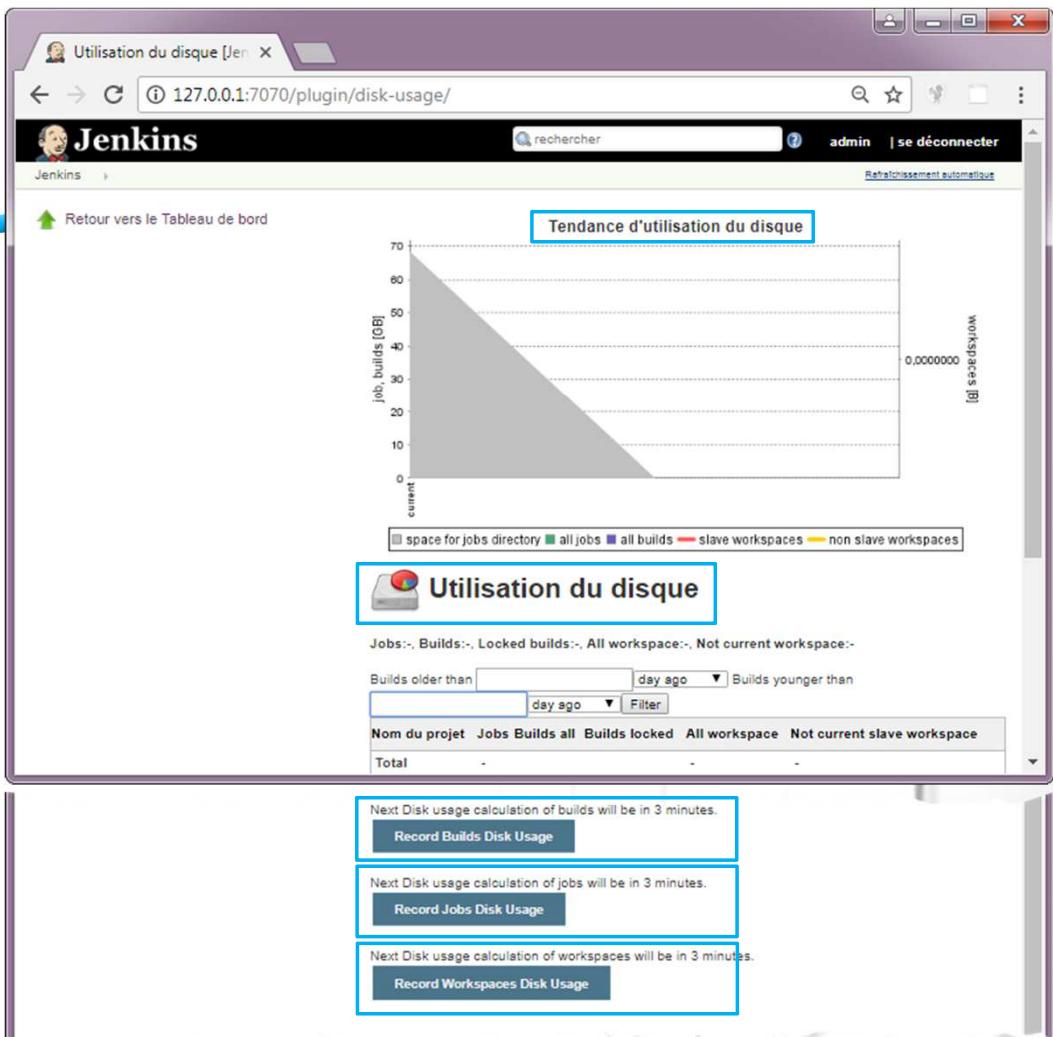


Le serveur d'intégration continue jenkins

23/08/2018

163

Gestion de l'espace disque



Le serveur d'intégration continue jenkins

Gestion de l'espace disque

- Le plugin Metrics Disk Usage affiche les détails de l'espace disque occupé par les jobs et workspaces des projets.
- Il affiche également la tendance d'utilisation du disque.
- L'utilisation du disque est calculée toutes les 60 minutes.
- Il est possible de planifier le calcul de l'utilisation du disque immédiatement.

Exercice

Installation du plugin Metrics Disk Usage

LAB

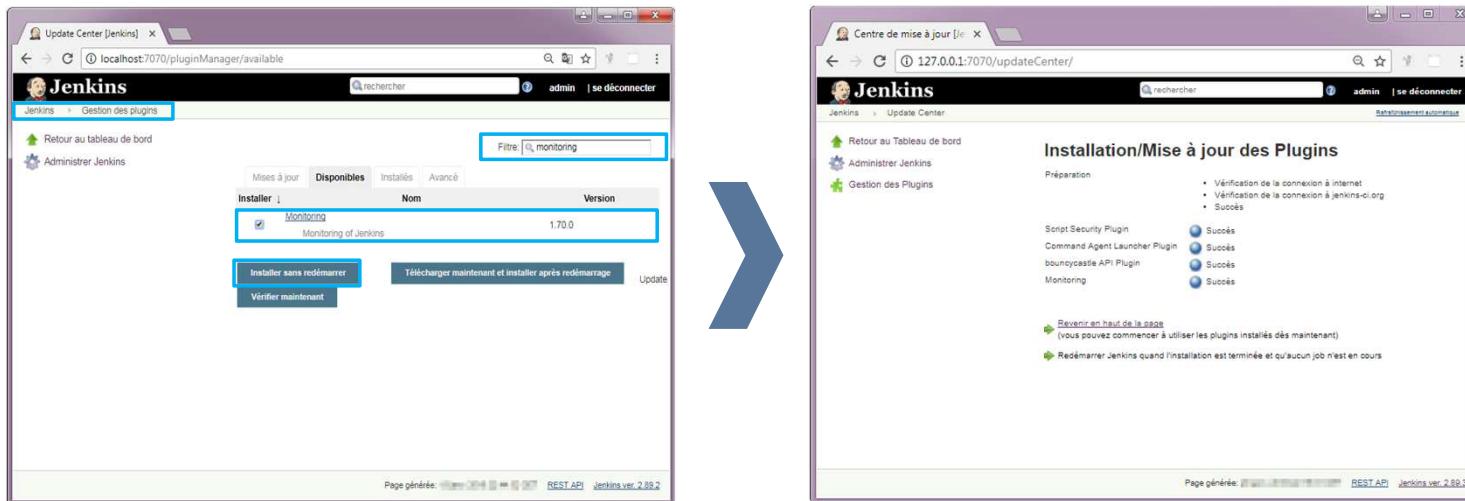
Le serveur d'intégration continue jenkins

Monitoring du serveur Jenkins

- Le plugin Monitoring fournit un ensemble d'information permettant d'assurer la surveillance du serveur Jenkins avec JavaMelody.
- Le plugin Monitoring fournit des statistiques et des états sur l'utilisation du processeur, de la mémoire, de la charge moyenne du système, du temps de réponse HTTP, etc.
- Il fournit également des informations détaillées sur les sessions HTTP, les erreurs et les journaux, l'activité du Garbage Collector, le vidage de tas, l'invalidation de sessions, etc.

Monitoring du serveur Jenkins

- L'installation du plugin Monitoring est disponible via le gestionnaire de plugin dans la vue « Administrer Jenkins > Gestion des plugins » dans l'onglet Disponible.



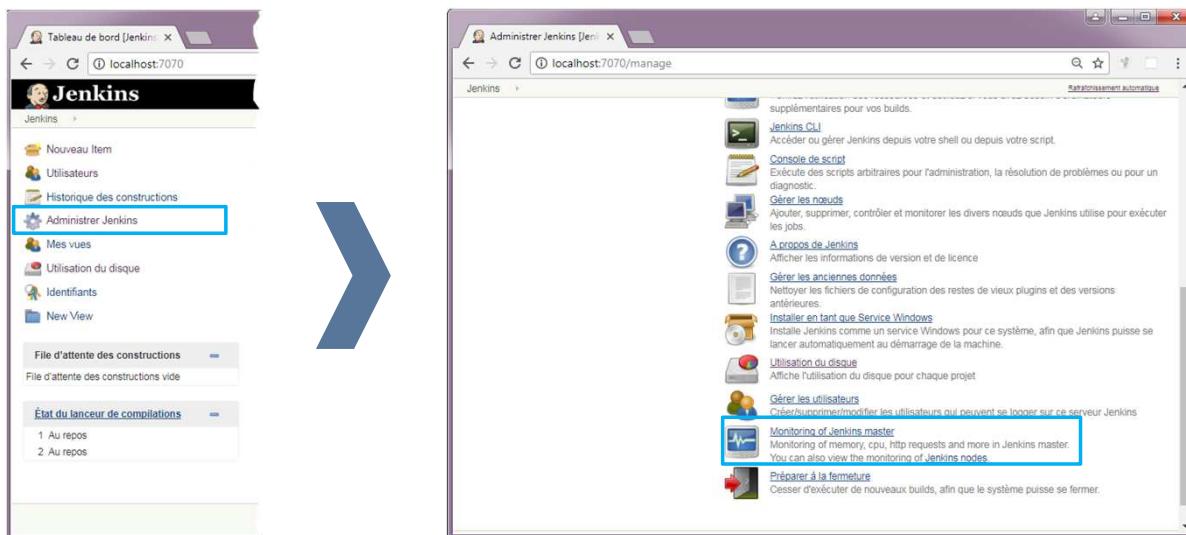
Le serveur d'intégration continue jenkins

23/08/2018

168

Monitoring du serveur Jenkins

- La vue de surveillance du serveur Jenkins est accessible via « Administre Jenkins > Monitoring of Jenkins master » après installation du plugin Monitoring.



Le serveur d'intégration continue jenkins

23/08/2018

169

Monitoring du serveur Jenkins

- La vue « Monitoring of Jenkins master » permet d'afficher les statistiques collectées par le plugin Monitoring.



Le serveur d'intégration continue jenkins

23/08/2018

170

Monitoring du serveur Jenkins

- Des informations détaillées sur les requêtes HTTP traitées par le serveur sont disponibles dans la rubrique « Statistiques http ».



Monitoring JavaMelody > Autres courbes

Statistiques http - 1 jour

| Requête | % du temps cumulé | Hits | Temps moyen (ms) | Temps max (ms) | Ecart-type | % du temps cpu cumulé | Temps cpu moyen (ms) | % d'erreur système | Taille moyenne (Ko) |
|-------------|-------------------|------|------------------|----------------|------------|-----------------------|----------------------|--------------------|---------------------|
| http global | 100 | 227 | 39 | 4 890 | 338 | 100 | 32 | 1,32 | 5 |
| http warnin | 12 | 2 | 573 | 944 | 524 | 13 | 499 | 0,00 | 18 |
| http error | 86 | 3 | 1 959 | 4 890 | 2 578 | 69 | 1 710 | 33,33 | 5 |

7 hits/min sur 76 requêtes ▾ Détails

| Requête | % du temps cumulé | Hits | Temps moyen (ms) | Temps max (ms) | Ecart-type | % du temps cpu cumulé | Temps cpu moyen (ms) | % d'erreur système | Taille moyenne (Ko) |
|---|-------------------|------|------------------|----------------|------------|-----------------------|----------------------|--------------------|---------------------|
| GET | 55 | 2 | 2 464 | 4 890 | 3 430 | 58 | 2 168 | 50,00 | 4 |
| /image GET | 12 | 2 | 573 | 944 | 524 | 13 | 499 | 0,00 | 18 |
| /login GET | 10 | 1 | 949 | 949 | 0 | 10 | 795 | 0,00 | 6 |
| /ajaxExecutors ajax POST | 4 | 46 | 8 | 89 | 14 | 2 | 4 | 0,00 | 0 |
| /updateCenter/body ajax POST | 3 | 59 | 5 | 38 | 5 | 3 | 4 | 0,00 | 1 |
| /lib/layout/menu_right_arrow.png GET | 2 | 1 | 254 | 254 | 0 | 3 | 234 | 100,00 | 2 |
| /ajaxBuildQueue ajax POST | 2 | 46 | 4 | 79 | 12 | 1 | 2 | 0,00 | 0 |
| Scripts/prototype.js GET | 1 | 1 | 169 | 169 | 0 | 0 | 62 | 0,00 | 161 |
| Sassets/jquery-detachedjsmodules/jquery2.js GET | 0 | 1 | 68 | 68 | 0 | 0 | 31 | 0,00 | 119 |
| Scripts/hudson-behavior.js GET | 0 | 1 | 44 | 44 | 0 | 0 | 31 | 0,00 | 105 |
| Scripts/yui/container/container-min.js GET | 0 | 1 | 34 | 34 | 0 | 0 | 15 | 0,00 | 74 |
| Scripts/yui/menu/menu-min.js GET | 0 | 1 | 25 | 25 | 0 | 0 | 15 | 0,00 | 56 |
| Scripts/yui/assets/skins/sam/skin.css GET | 0 | 1 | 24 | 24 | 0 | 0 | 31 | 0,00 | 107 |
| Scss/style.css GET | 0 | 1 | 23 | 23 | 0 | 0 | 31 | 0,00 | 37 |
| Scripts/yui/animation/animation-min.js GET | 0 | 1 | 18 | 18 | 0 | 0 | 15 | 0,00 | 13 |
| Scripts/yui/button/button-min.js GET | 0 | 1 | 17 | 17 | 0 | 0 | 15 | 0,00 | 30 |
| Sjbundles/page-init.js GET | 0 | 1 | 17 | 17 | 0 | 0 | 0 | 0,00 | 37 |
| Scripts/yui/event/event-min.js GET | 0 | 1 | 16 | 16 | 0 | 0 | 15 | 0,00 | 14 |
| Scss/responsive-grid.css GET | 0 | 1 | 15 | 15 | 0 | 0 | 0 | 0,00 | 29 |
| Scripts/yui/autocomplete/autocomplete-min.js GET | 0 | 1 | 14 | 14 | 0 | 0 | 15 | 0,00 | 31 |
| /administrativeMonitor/hudson/diagnosis/ReverseProxySetupMonitor/test ajax POST | 0 | 2 | 5 | 9 | 4 | 0 | 7 | 0,00 | 0 |
| /lib/layout/menu_right_arrow2.png GET | 0 | 1 | 11 | 11 | 0 | 0 | 0 | 0,00 | 2 |
| Scripts/yui/vahoo/vahoo-min.js GET | 0 | 1 | 10 | 10 | 0 | 0 | 0 | 0,00 | 8 |

Le serveur d'intégration continue jenkins

Monitoring du serveur Jenkins

- Les erreurs systèmes sont reportées dans les rubriques « Statistiques erreurs systèmes http » et « Statistiques logs d'erreurs systèmes.

The screenshot shows a Java application window titled "Monitoring JavaMelody" with the URL "127.0.0.1:7070/monitoring". The main content area is divided into two sections: "Statistiques erreurs systèmes - 1 jour" and "Statistiques logs d'erreurs systèmes - 1 jour".

Statistiques erreurs systèmes - 1 jour:

| Erreur | % du temps cumulé | Hits | Temps moyen (ms) | Temps max (ms) | Ecart-type | % du temps cpu cumulé | Temps cpu moyen (ms) |
|---|-------------------|------|------------------|----------------|------------|-----------------------|----------------------|
| hudson.security.AccessDeniedException2: anonymous is missing the Global/Read permission | 13 | 2 | 19 | 38 | 26 | 6 | 7 |
| org.eclipse.jetty.io.EofException | 86 | 1 | 254 | 254 | 0 | 93 | 234 |

Statistiques logs d'erreurs systèmes - 1 jour:

| Erreur | Hits |
|---------------|------|
| WARNING: null | 2 |

Logs d'erreurs (Partial Data):

| Date | Requête | Erreur |
|-------------------|--|---|
| 20/01/18 22:22:50 | GET / | hudson.security.AccessDeniedException2: anonymous is missing the Global/Read permission |
| 20/01/18 22:22:51 | /adjuncts/7da99139/lib/layout/menu_right_arrow.png | GET org.eclipse.jetty.io.EofException |
| 20/01/18 22:22:59 | /opensearch.xml | GET hudson.security.AccessDeniedException2: anonymous is missing the Global/Read permission |

Requêtes en cours:
Aucune requête en cours

Exercice

Installation du plugin Monitoring

LAB

Le serveur d'intégration continue jenkins

23/08/2018

173

Création de Job Jenkins

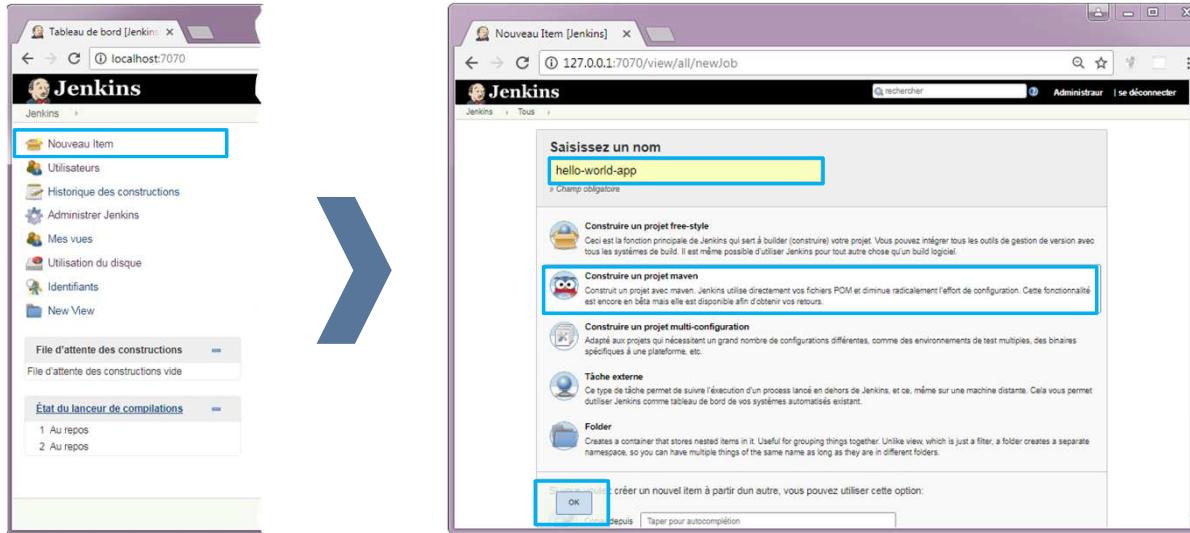
- La construction d'un projet est assurée par un Job dans Jenkins.
- La configuration du Job détermine comment le projet sera construit:
 - Comment compiler le projet.
 - Comment tester le projet.
 - Comment empaqueter le projet.
 - Comment déployer le projet.
 - Etc.
- La configuration d'un Job comporte les éléments suivants:
 - La configuration de la gestion de code source.
 - La configuration des éléments déclencheurs de la construction du projet.

Création de Job Jenkins

- La configuration de l'environnement de la construction du projet.
- Les étapes à dérouler avant de lancer la construction du projet.
- La configuration de l'outil de construction du projet.
- Les étapes à dérouler après la fin de la construction du projet.

Création de Job Jenkins

- La création de nouveau Job Jenkins est disponible via la vue « Nouveau Item ».



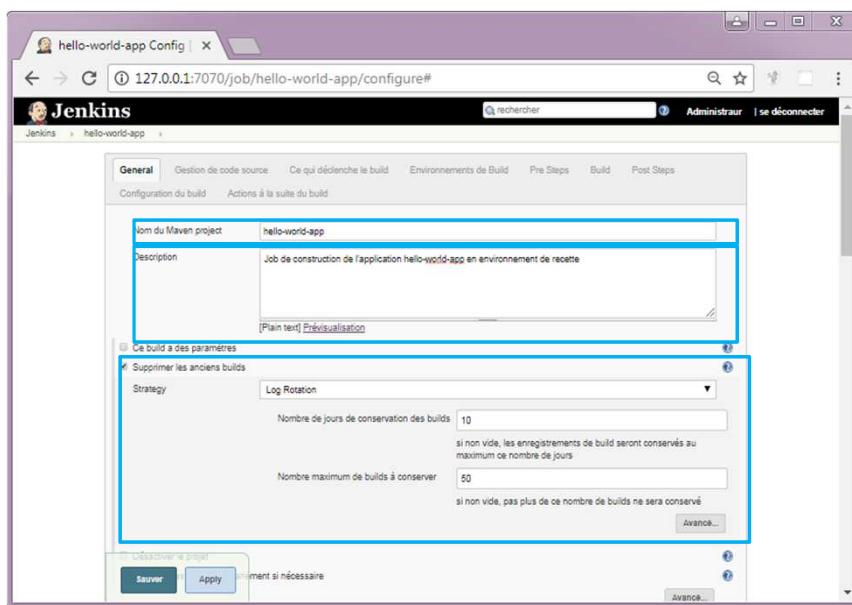
Le serveur d'intégration continue jenkins

23/08/2018

176

Création de Job Jenkins

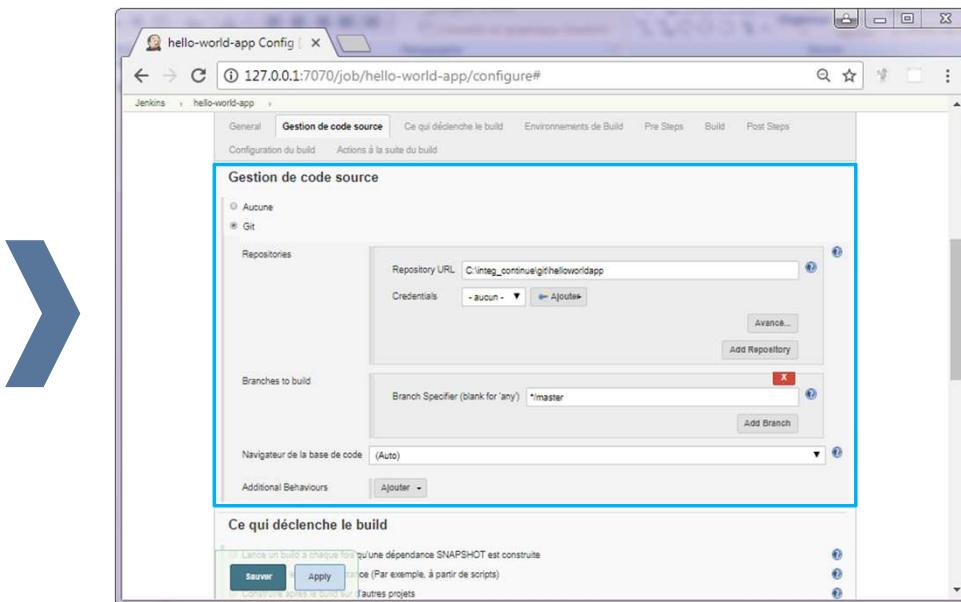
- La section « General » permet de configurer des informations générales sur le projet, telle que son nom et une description, le nombre de construction à conserver en historique.



Le serveur d'intégration continue jenkins

Création de Job Jenkins

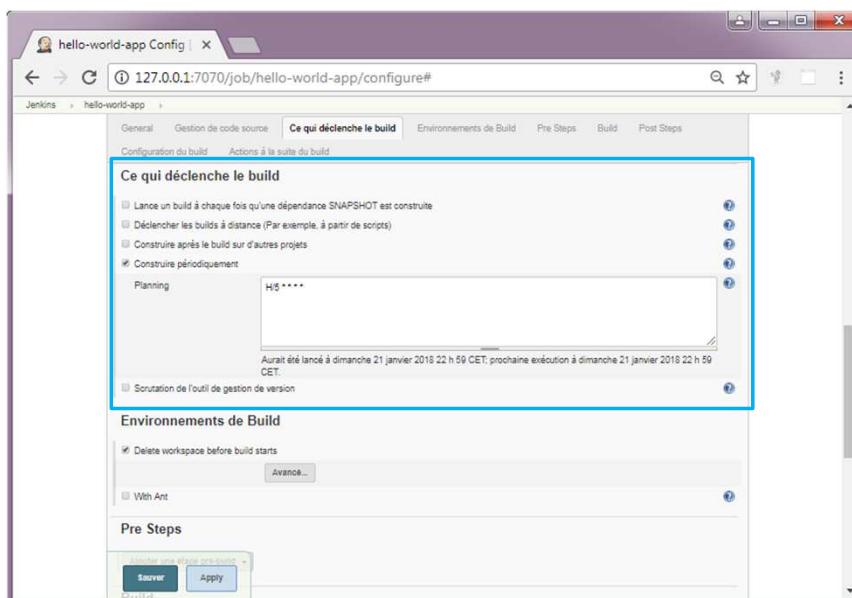
- La section « Gestion de code source » permet de configurer l'accès au dépôt Git contenant le code source du projet.



Le serveur d'intégration continue jenkins

Création de Job Jenkins

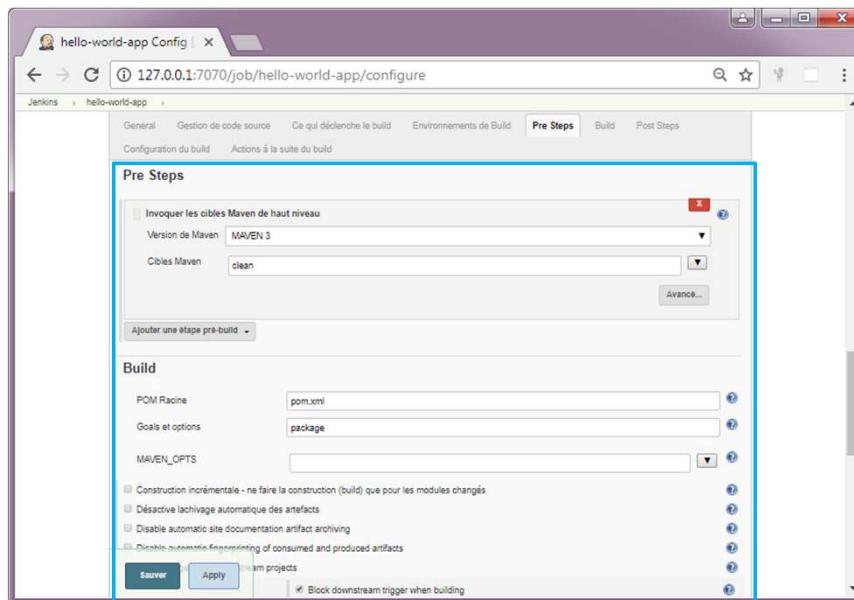
- La section « Ce qui déclenche le build » permet de configurer l'élément déclencheur de construction.
 - Une construction est programmée toutes les 5 minutes.



Le serveur d'intégration continue jenkins

Création de Job Jenkins

- Les sections « Pre Steps » et « Build » permettent de configurer les étapes de construction Maven. Indiquez le nom du fichier de construction.



Le serveur d'intégration continue jenkins

Création de Job Jenkins

- Sauvegarder la configuration du Job et cliquer sur le bouton « Lancer un build » pour lancer la construction du projet immédiatement.
- L'avancement de la construction du projet est visible dans la widget « Historique des builds ».



The screenshot shows the Jenkins interface for the 'hello-world-app' job. On the left, there's a sidebar with options like 'Retour au tableau de bord', 'État', 'Modifications', 'Répertoire de travail', 'Lancer un build' (which is highlighted with a blue box), 'Supprimer Maven project', 'Configurer', and 'Modules'. The main content area is titled 'Maven project hello-world-app' and describes it as a 'Job de construction de l'application hello-world-app en environnement de recette'. It includes links for 'Espace de travail', 'Changements récents', and 'Liens permanents' (with a link to the last build). At the bottom, there's a 'Historique des builds' section with a table showing the last build (#1), which was run 1.5s ago. There are also 'RSS des builds' and 'RSS des échecs' links. The bottom right corner of the window shows 'Page générée le 23/08/2018 à 14:59:23 REST API Jenkins ver. 2.99.3'.

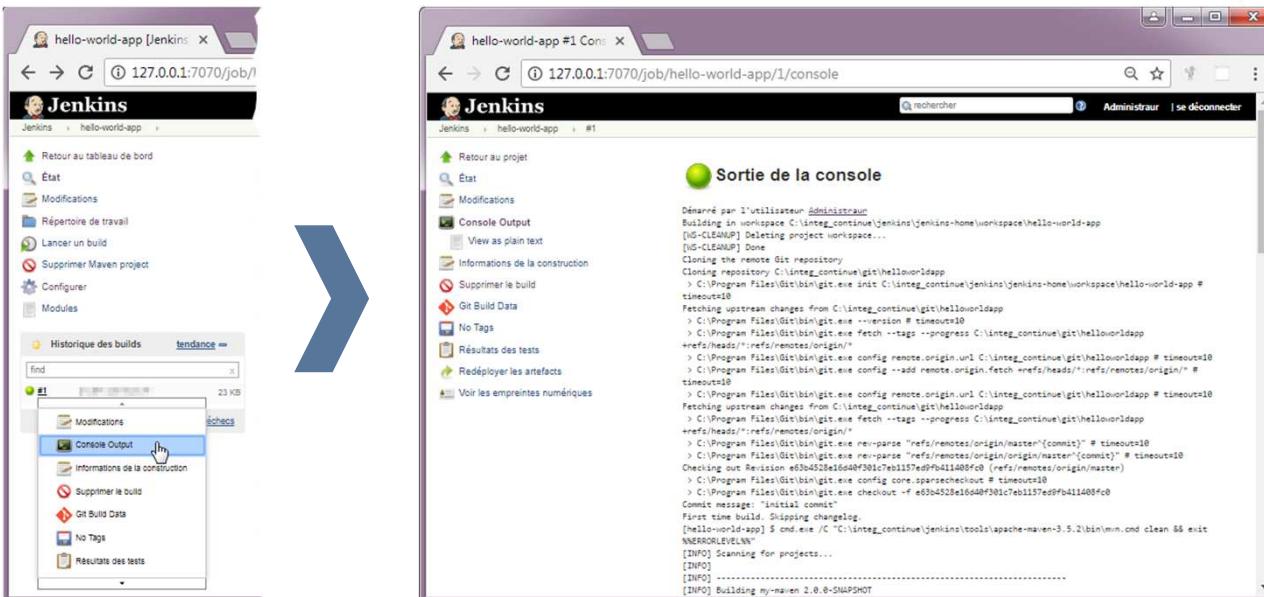
Le serveur d'intégration continue jenkins

23/08/2018

181

Création de Job Jenkins

- Les logs produits lors de l'exécution de la construction du projet sont accessibles via la sortie de console.



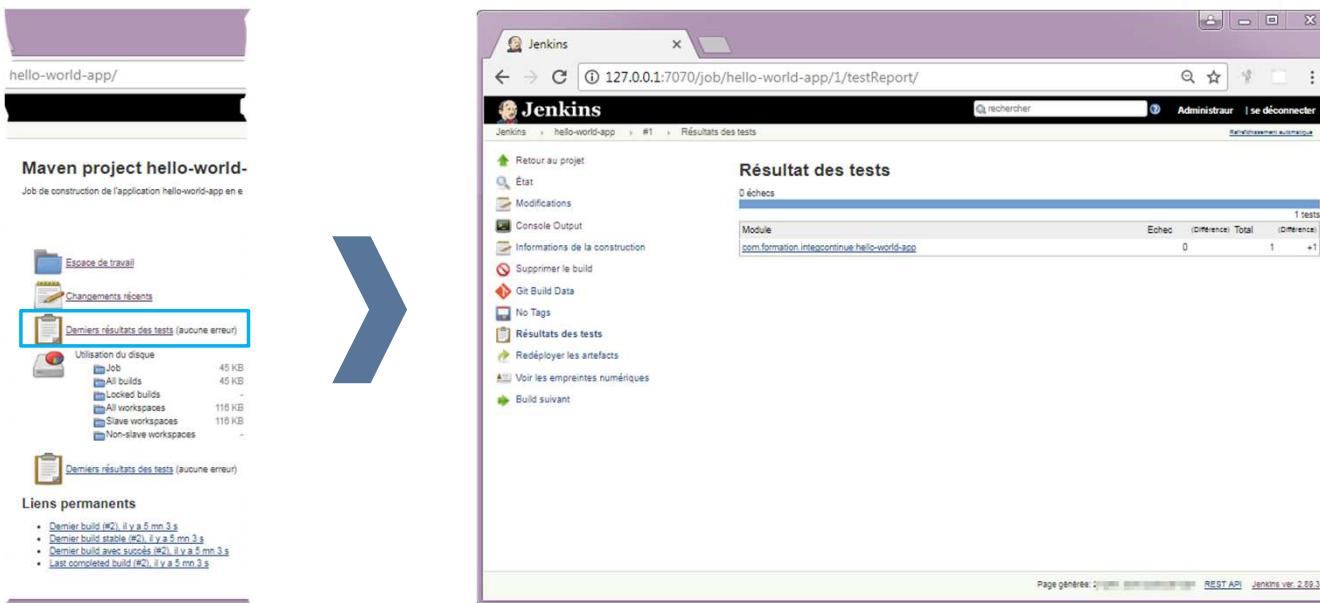
Le serveur d'intégration continue jenkins

23/08/2018

182

Création de Job Jenkins

- Les résultats de l'exécution des tests unitaires sont accessible via le lien « Derniers résultats des tests ».



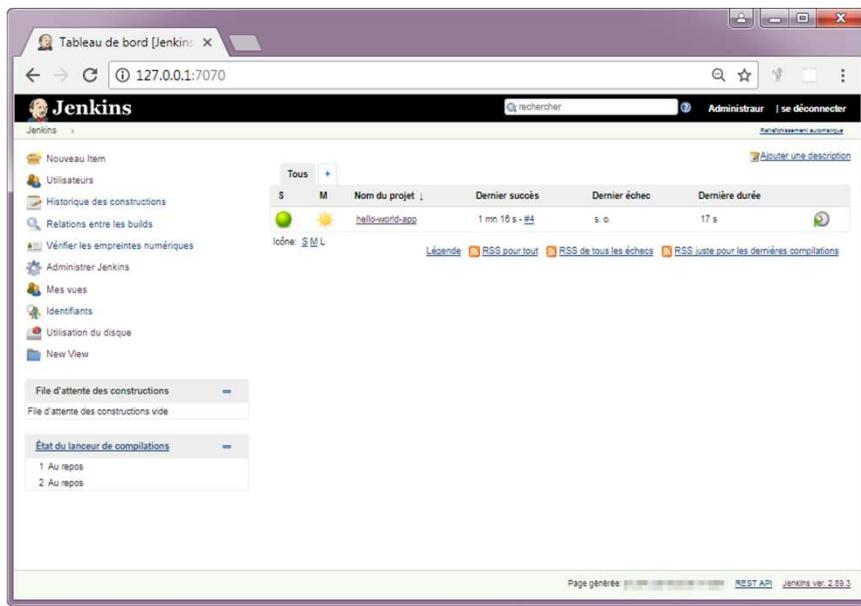
Le serveur d'intégration continue jenkins

23/08/2018

183

Création de Job Jenkins

- Le tableau de bord Jenkins affiche le statut de la dernière construction du projet ainsi que le niveau de stabilité des dernières constructions.
- Le tableau de bord Jenkins fournie un raccourcis pour lancer la construction d'un projet.



Le serveur d'intégration continue jenkins

23/08/2018

184

Sommaire

- **Chapitre 1 : Introduction**
- **Chapitre 2 : L'intégration Continue**
- **Chapitre 3 : Automatisation des Builds**
- **Chapitre 4 : Le gestionnaire de sources**
- **Chapitre 5 : Le serveur d'intégration continue : Jenkins**
- **Chapitre 6 : La gestion des livrables**

Gestion des livrables

- Agenda
 - Notion de dépôt d'artefacts
 - Stratégie de mise à disposition du résultat construit
 - Le gestionnaire de dépôt d'artefacts Nexus

Notion de dépôt d'artefacts

- Un dépôt d'artefacts, dit également repository, est une collection d'artefacts logiciels binaires et de métadonnées stockés dans une structure de répertoire définie, qui est utilisée par des clients tels que Maven pour récupérer des binaires.
- Dans le cas des dépôts d'artefacts Maven, le type principal d'artefact binaire est un fichier JAR.
- Il n'y a pas de limite au type d'artefacts pouvant être stockés dans un dépôt Maven qui peut stocker également:
 - Des archives de documentation.
 - Des archives de code sources, etc.

Notion de dépôt d'artefacts

- Un dépôt fournit une plate-forme pour le stockage, la récupération et la gestion des artefacts et des métadonnées du logiciel binaire.
- Dans le paradigme Maven, chaque artefact logiciel est décrit par un document XML (Project Object Model ou POM).
- Le POM contient des informations qui décrivent un artefact logiciel et répertorient ses dépendances, ie les artefacts logiciels binaires dont dépend un composant donné pour une compilation ou une exécution réussie.
- Lorsque Maven télécharge une dépendance à partir d'un dépôt d'artefacts, il télécharge également le POM de cette dépendance.

Stratégie de mise à disposition du résultat : Dépôts de Release et dépôts de Snapshot

- Les dépôts de release sont destinés aux artefacts de release stables.
- Les dépôts de snapshot sont des dépôts fréquemment mis à jour, utilisés pour le stockage des artefacts logiciels provenant de projets en cours de développement.
- Les dépôts ont généralement segmentés en release ou snapshot servant différents consommateurs et appliquant des normes et des procédures différentes pour le déploiement d'artefacts.
- Un dépôt de release est considéré comme un dépôt de production et un dépôt de snapshot est considéré comme un dépôt de développement ou de test.

Stratégie de mise à disposition du résultat : Dépôts de Release et dépôts de Snapshot

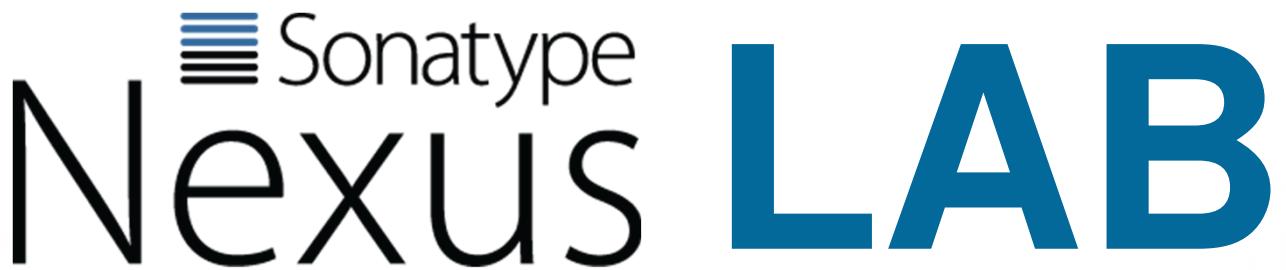
- Les artefacts snapshot sont des artefacts générés lors du développement d'un projet logiciel.
- Les artefacts snapshot peuvent être déployés et modifiés fréquemment sans tenir compte des problèmes de stabilité et de répétabilité.

Les gestionnaires de dépôt d'artefacts

- Un gestionnaire de dépôt d'artefacts est:
 - Un proxy pour les dépôts distants qui met en cache les artefacts, en économisant à la fois la bande passante et le temps requis pour récupérer un artefact logiciel à partir d'un dépôt distant.
 - Un hôte pour les artefacts internes fournissant à une organisation un environnement de déploiement pour les artefacts logiciels.
- Un gestionnaire de dépôt d'artefacts est capable de gérer les artefacts logiciels binaires.
- Un gestionnaire de dépôt d'artefacts peut rechercher des artefacts logiciels, vérifier des opérations de développement et de publication.

Exercice

Installation d'un serveur Nexus



Le serveur d'intégration continue jenkins

Questions & Réponses

Q&R

GESTION DES LIVRABLES

23/08/2018

193